ROOLIPOHJAINEN KÄYTTÖOIKEUKSIEN HALLINTA

LAHDEN AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2010
Kimmo Snellman

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

SNELLMAN, KIMMO: Roolipohjainen käyttöoikeuksien hallinta

Ohjelmistotekniikan opinnäytetyö, 75 sivua, 4 liitesivua

Kevät 2010

TIIVISTELMÄ

Tässä opinnäytetyössä tutkitaan roolipohjaista käyttöoikeuksien hallintaa käyttöoikeuksien hallintajärjestelmissä. Käyttöoikeudella tarkoitetaan mitä tahansa henkilön työssään tarvitsemaa tietojärjestelmää tai fyysistä laitetta, jonka käyttämiseksi tarvitaan lupa. Rooli on vastaavasti joukko käyttöoikeuksia, joita henkilö tarvitsee suoriutuakseen työtehtävistään. Roolipohjaisessa käyttöoikeuksien hallinnassa käyttöoikeuksia haetaan, myönnetään tai poistetaan kerralla roolien avulla sen sijaan, että käyttöoikeuksia hallittaisiin yksittäisinä.

Työn tavoitteena on selvittää, mitä roolipohjaisella käyttöoikeuksien hallinnalla tarkoitetaan ja mitkä tekijät vaikuttavat siihen, millainen roolipohjaisen käyttöoikeuksien hallinta sopii erilaisille organisaatioille. Työssä kerrotaan yleisesti käyttöoikeuksien hallinnan aihepiiristä ja syvennytään tarkemmin roolipohjaisen käyttöoikeuksien hallinnan ominaispiirteisiin ja käsitteisiin. Lisäksi perehdytään aihepiirin keskeisimpiin lakeihin ja säädöksiin, standardeihin, menetelmiin ja malleihin. Työn lopputuloksena syntyvää tietoa voidaan käyttää apuna perehtyessä aihealueeseen ennen käyttöoikeuksien hallintaprojektin aloittamista, laajennettaessa olemassa olevaa käyttöoikeuksien hallintaa roolipohjaisuuteen tai suunniteltaessa roolipohjaisia käyttöoikeuksien hallintasovelluksia.

Tutkimuksen asiakastapauksena toimii tietojärjestelmäprojekti, jossa Salon kaupungille toimitettiin roolipohjainen käyttöoikeuksien hallintajärjestelmä. Salon kaupunki tavoitteli projektissa keskitettyä käyttöoikeuksien hallinnan prosessia ja järjestelmää, jonka avulla se pystyisi hallitsemaan roolipohjaisesti kaikkia kaupungin sekä terveydenhuollon työntekijöiden käyttöoikeuksia.

Tutkimuksen lopputuloksena selviää, mitä varten roolipohjainen käyttöoikeuksien hallinta on luotu ja mitä käsitteellä rooli tarkoitetaan eri asiayhteyksissä. Rooleista puhuttaessa on tärkeää erottaa työ- ja järjestelmäroolien merkitys. Roolien avulla pyritään tuomaan hallittavuutta, nopeutta, johdonmukaisuutta, kustannussäästöjä ja tietoturvallisuutta käyttöoikeuksien hallintaan. Tavoitteisiin pääsemiseksi on huomioitava on kaikki toimintaympäristöön liittyvät tekijät ja arvioitava tapauskohtaisesti oikeat menetelmät roolien määrittämiseksi. Käyttäjien määrä, hallittavien käyttöoikeuksien määrä, tietojärjestelmien laajuus, organisaatiorakenne, liiketoimintaan ja toimialaan liittyvät ominaispiirteet, säädökset ja tietoturvavaatimukset vaikuttavat kaikki oikean toimintatavan valintaan.

Avainsanat: käyttöoikeuksien hallinta, RBAC, IDM, IAM, roolit

Lahti University of Applied Sciences
Degree Programme in Information Technology

SNELLMAN, KIMMO: Role-based access rights management

Bachelor's Thesis in Software Engineering, 75 pages, 4 appendices

Spring 2010

ABSTRACT

This study deals with role-based access rights management in access rights management systems. Access right here means any information system or physical device to which a person needs a permission, to use it in their job. In role-based access rights management systems access rights are requested, approved or removed by using roles rather than managing access rights separately.

The aim of this study was to find out the meaning of the role-based access rights management and the factors which influence what kind of role-based access rights management suits different organizations. First there is a general description of access rights management and then the focus is on the concept of role-based access control. After that common laws, standards, models and methods of role-based access rights management are described. The results of the study can be used for getting familiar with the subject before an identity management project, changing current system to role-based access control or planning the role-based identity management software.

The customer case in this study is a project where role-based access rights management software was delivered to the city of Salo. The aim of the city was to create a centralized process for handling access rights. The project included the identity management software which can be used to handle all the access rights of the personnel of the city and its healthcare service.

The study shows why the role-based access rights management was created and what the definition of role means in different contexts. It is very important to separate the meaning of the task roles and system roles. By using roles, access right management can be more manageable, faster, more consistent, more cost-effective and more secure. To achieve these goals all factors related to the operational environment have to be taken into account and the right methods for defining roles must be chosen. The number of users, the number of access rights, the size of system environment, the organization structure, the special characteristics of the business area, laws and demands for security do have an effect on how to choose the right way to use role-based access rights management.

Keywords: access rights management, RBAC, role-based access control, IDM, IAM, roles

SISÄLLYS

SANASTO

ARBAC	Adminitrative Role-based Access Control. Roolipohjaisen pääsynhallinnan mallin laajennus, jossa RBAC-mallia käytetään itsessään roolien ylläpitoon.
Bottom-up-menetelmä	Roolien määrittämisen menetelmä, jossa rooleja tutkitaan perustuen olemassa oleviin käyttöoikeustietoihin. (=Role-mining)
DAC	Discretionary Access Control. RBAC-mallia edeltävä pääsynhallinnan malli, jossa käyttöoikeuksien myöntäminen perustuu harkintaan.
ERBAC	Enterprise Role-Based Access Control. Laajennus RBAC-standardista, jolla hallitaan työrooleja.
HR-järjestelmä	Human Resources. Henkilöstötietojen hallintajärjestelmä.
IDM, Identity Management	Identiteettien hallinta. Käytetään myös synonyymina käyttöoikeuksien tai käyttövaltuuksien hallinnalle.
Järjestelmärooli	Tietojärjestelmien sisäisten oikeuksien hallintaan tarkoitettu rooli.
LDAP	Lightweight Directory Access Protocol. TCP/IP-pohjainen hakemistopalvelujen saantikäytäntö.
Least priviledges	Vähäisimpien käyttöoikeuksien periaate, jonka mukaan henkilöllä tulisi olla vain ne käyttöoikeudet, mitä hän välttämättä työssään tarvitsee.
MAC	Mandatory Access Control. RBAC-mallia edeltävä pääsynhallinnan malli, jossa käyttöoikeuksien myöntäminen perustuu tietosisältöön ja turvallisuustasoon
Provisiointi	Käyttäjä- ja käyttövaltuustietojen välittäminen/luominen tietojärjestelmiin.
RBAC	Role-based Access Control. Roolien mukainen käyttöoikeuksien hallinta.
Role engineering	Roolien määrittelyä tarkoittava termi.
Role-mining	Roolien tunnistamisen menetelmä, jossa rooleja tutkitaan perustuen olemassa oleviin käyttöoikeustietoihin. (=Bottom-up-menetelmä)
Separation of Duties	Roolien määrittelyn periaate, jolla rajataan tietoturvan kannalta vaarallisia käyttöoikeusyhdistelmiä.
SOX	Sarbanes-Oxley Act. Yhdysvaltalainen laki (2002), joka asettaa määräyksiä kaikkien Yhdysvalloissa pörssinoteerattujen yritysten hallintoon, johtamiseen

	sekä tilintarkastukseen sekä käyttöoikeuksien hallintaan.
SPML	Service Provisioning Markup Language. XML-pohjaisia standardeja kehittävän OASIS-standardointijärjestön standardi, joka kuvaa palvelumääritysten siirrossa järjestelmästä toiseen sovellettavia käytäntöjä.
Top-down-menetelmä	Roolien määrittämisen menetelmä, jossa rooleja tutkitaan perustuen prosessikuvauksiin ja organisaatiokaavioihin.
Työrooli	Työtehtävien suorittamissa tarvittavien käyttöoikeuksien hallintaan tarkoitettu rooli.
Workflow	Työvuo, työnkulku.

1 JOHDANTO

Organisaatioiden ja yritysten henkilöt tarvitsevat tietyn määrän käyttöoikeuksia, joiden avulla he pystyvät suoriutumaan tehtävistään. Käyttöoikeuksilla voidaan tarkoittaa tietojärjestelmiä, joiden käyttämiseksi vaaditaan tunnus ja salasana, oikeutta käyttää toimistosovellusta tai oikeutta fyysiseen laitteeseen, kuten esimerkiksi puhelimeen tai kulkulupaan. Käyttöoikeus on mikä tahansa asia, minkä käyttämiseksi tarvitaan lupa. Yrityksissä ja organisaatioissa käyttöoikeuksia ovat perinteisesti myöntäneet tekniset henkilöt tai mikrotuki, tai tehtävät voivat olla hajautettuna eri järjestelmien pääkäyttäjille tai vastuuhenkilöille. Käyttöoikeuksia on yrityksissä ja organisaatioissa vaikea kontrolloida, ja niitä hallitaan usein manuaalisesti. Käyttöoikeuksia jaetaan helposti liikaa, minkä seurauksena tietoturvariskit ja väärinkäytöksien mahdollisuudet kasvavat.

Käyttöoikeuksien hallinnan parantamiseksi on tehty hallintajärjestelmiä, joiden avulla voidaan keskitetysti hallita käyttöoikeusprosessia työntekijöiden koko elinkaaren ajan. Järjestelmien avulla voidaan anoa henkilöille tarvittavat käyttöoikeudet työsuhteen alkaessa tai sen aikana ja hyväksyttää oikeudet esimiehillä ennen käyttöoikeuksien myöntämistä. Järjestelmiin jää kaikista tapahtumista lokimerkinnät, jotka mahdollistavat raportoinnin ja seurattavuuden. Käyttöoikeuksien hallitseminen järjestelmistä huolimatta on sitä vaikeampaa, mitä isoimmista organisaatioista on kyse. Isoilla organisaatioilla voi olla hallittavanaan satoja eri järjestelmiä tai fyysisiä laitteita, joihin voi liittyä tuhansia käyttöoikeuksia.

Käyttöoikeuksien hallinnan tehostamiseksi on luotu roolipohjainen käyttöoikeuksien hallinnan periaate, jonka mukaan käyttöoikeuksia ei myönnetä henkilöille yksittäisinä vaan tietyn työnkuvaan tai tehtävään liittyvän työroolin kautta. Yksinkertaistettuna työrooli on joukko käyttöoikeuksia, joita henkilö tarvitsee suoriutuakseen työtehtävistään. Roolien avulla voidaan myös hallita yksittäisen järjestelmän pääsynhallintaa, jolloin rooleilla tarkoitetaan RBAC-standardin mukaisia järjestelmärooleja.

Työn ensisijaisena tavoitteena on selvittää, mitä roolipohjaisella käyttöoikeuksien hallinnalla tarkoitetaan. Lisäksi tavoitteena on selvittää, mitkä tekijät vaikuttavat siihen ja millainen roolipohjaisen käyttöoikeuksien hallinta sopii erilaisille organisaatioille. Työssä tutkitaan, mitä roolit tarkoittavat, millaisia erilaisia käyttöoikeuksien hallintaa liittyviä rooleja on olemassa ja millaisin menetelmin rooleja voidaan määritellä. Teoriaosuudessa selvitetään, mitä standardeja ja malleja on olemassa liittyen rooleihin ja roolien määrittelyn menetelmiin ja mihin standardit ja mallit on tarkoitettu.

Työn lopputuloksena syntyy tiivis tietopaketti roolipohjaisesta käyttöoikeuksien hallinnasta. Tietoa voidaan käyttää apuna perehtyessä aihealueeseen ennen käyttöoikeuksien hallintaprojektin aloittamista, laajennettaessa olemassa olevaa käyttöoikeuksien hallintaa roolipohjaisuuteen tai suunniteltaessa käyttöoikeuksien hallintasovelluksia. Työssä pohditaan ja tutkitaan roolipohjaista käyttöoikeuksien hallintaa erityisesti käyttöönottoprojektien ja käyttöoikeusprosessien näkökulmasta. Työssä pyritään selkeyttämään projektihenkilöille, projektipäälliköille, loppukäyttäjille ja asiakkaille, mitä aihepiirillä tarkoitetaan ja mitä tulee huomioida, kun suunnitellaan roolipohjaisen käyttöoikeuksien hallintasovelluksen käyttöönottoa.

Roolipohjaisten käyttöoikeuksien hallintasovellusten käyttöönottoprojektit epäonnistuvat usein. Lopputuloksena ei saavuteta parempaa käyttöoikeushallinnan prosessia, rooleja syntyy määrittelyprojektin seurauksena yhtä paljon kuin yrityksessä on käyttäjiä, käyttöönottoprojektin aikataulu ylittyy tai muuttuvasta ympäristöstä johtuen roolien ylläpito on kalliimpaa kuin ilman rooleja toimiva käyttöoikeuksien hallinta.

Onkin syytä arvioida, miksi näin tapahtuu. Teoriassa roolipohjaisuus tuo käyttöoikeuksien hallintaan useita etuja: nopeutta, joustavuutta, hallittavuutta ja tietoturvaa. Työn tutkimusongelmina on selvittää, miksi teoria ei kohtaa reaalimaailmaa roolipohjaisessa käyttöoikeuksien hallinnassa ja mitkä seikat tekevät roolipohjaisista käyttöoikeuksien hallintaprojekteista vaikeita toteuttaa.

RBAC-standardit (eng. Role-based Access Control) keskittyvät yksittäisen järjestelmän näkökulmasta roolipohjaiseen käyttöoikeuksien hallintaan, vaikka samaa periaatetta voidaan käyttää järjestelmäriippumattomien työroolien kanssa. Tässä tutkimuksessa tutustutaan keskeisimpiin RBAC-standardeihin ja malleihin, mutta ei keskitytä yksityiskohtaisesti yksittäisen järjestelmän roolipohjaiseen pääsynhallintaan. Tutkimuksessa keskitytään kokonaisvaltaiseen roolipohjaiseen käyttöoikeuksien hallintaan.

Työssä käydään läpi roolien määrittelyyn liittyvät käsitteet ja menetelmät. Roolien määrittelyn automatisointiin on myös tehty erilaisia kaupallisia sovelluksia, jotka perustuvat erilaisiin laskentamenetelmiin ja algoritmeihin. Näihin ns. Role-mining sovelluksiin ei tässä tutkimuksessa perehdytä tarkemmin.

Tutkimuksessa tuodaan tietoutta myös ohjelmistokehittäjille, joiden tehtäviin kuuluu suunnitella roolipohjaisia käyttöoikeuksien hallintasovelluksia tai sovelluksia, joiden pääsynhallinta toteutetaan roolipohjaisten RBAC-standardien mukaisesti. Työn tarkoitus ei ole kuitenkaan toimia kattavana teknisenä ohjeena.

Tutkimuksen asiakastapauksena toimi vuonna 2008 joulukuussa alkanut tietojärjestelmäprojekti, jossa oli tarkoitus toimittaa Salon kaupungille roolipohjainen käyttöoikeuksien hallintajärjestelmä ja luoda järjestelmän avulla keskitetty käyttöoikeuksien hallinnan prosessi. Käyttöoikeuksien hallinta kohdistuu noin viiteen tuhanteen kaupungin sekä terveydenhuollon työntekijään.

Järjestelmän käyttöönotto oli vielä meneillään tutkimuksen tekemisen aikana vuoden 2010 alussa. Järjestelmätoimittajana oli Propentus Oy, jossa tutkimuksen tekijä toimi projektipäällikkönä ja vastuullisena henkilönä kyseisessä projektissa. Koska Salon projekti oli jo alkanut ennen tutkimuksen tekemistä, ei asiakastapausta ole käytetty tämän tutkimuksen kohteena suoraan vaan lähinnä esimerkkitapauksena. Tutkimuksen teoriaosuutta peilataan projektin käytännön kokemuksiin ja arvioidaan, kuinka teorioita on hyödynnetty projektissa.

2 KÄYTTÖOIKEUKSIEN HALLINTA

2.1 Yleistä

Käyttöoikeuksien hallinnan lähtökohtana on periaate, jonka mukaan henkilöllä tulee olla työtehtävien suorittamisessa tarvittavat työkalut. Toisaalta henkilöllä ei pidä olla käytössään työkaluja, mitä työssä ei tarvita ja mistä voi aiheutua kustannuksia tai tietoturvariskejä työnantajalle. Työkaluilla tarkoitetaan tässä yhteydessä käyttöoikeuksia. Käyttöoikeuksiksi mielletään yleisesti henkilöiden verkkotunnukset, sähköpostiosoitteet tai käyttöoikeudet tietojärjestelmiin, joiden käyttämiseksi vaaditaan käyttäjätunnukset. Näiden lisäksi käyttöoikeus voi kohdistua toimisto-ohjelmiin tai fyysisiin laitteisiin, kuten puhelimiin, tietokoneisiin tai kulkulupiin. Käyttöoikeudella tässä tutkimuksessa tarkoitetaan mitä tahansa järjestelmää, sovellusta, laitetta tai asiaa, johon tulee myöntää käyttöoikeus. (Propentus 2009, 4-6.)

Käyttöoikeuksien hallinnan tavoitteena on luoda kontrolloitu prosessi käyttöoikeuksien myöntämiseen ja poistamiseen. Hallintaan kuuluu myös raportointi, jonka avulla pystytään selvittämään ja seuraamaan, mitä käyttöoikeuksia kenelläkin on, kuka nämä oikeudet on myöntänyt ja milloin oikeudet on myönnetty. Kontrolloidun prosessin ja prosessia tukevan tietojärjestelmän avulla voidaan parantaa tietoturvaa, vähentää käyttöoikeuksien hallinnan manuaalista rutiinityötä, vähentää lisenssikustannuksia, lisätä henkilöiden tietoisuutta käyttöoikeuksista sekä selkeyttää käyttöoikeuksien hallinnan vastuita ja velvollisuuksia. (Propentus 2009, 4-6.)

Kehnosti hoidettu käyttöoikeuksien hallinta aiheuttaa yrityksille lukuisia ongelmia. Yritykset eivät hallitse käyttöoikeuksiaan, tietoturvan taso on huono, henkilöille jää tunnuksia ja salasanoja voimaan, vaikka työsuhde on päättynyt. Lisenssikulut ovat korkeat, kun henkilöillä on käytössään sovelluksia, mitä he eivät työtehtävissään tarvitse. Käyttöoikeuksien hallintaan kuluu runsaasti manuaalista rutiinityötä. Isoissa organisaatioissa, joissa on useita kymmeniä järjestelmiä ja

tuhansia käyttäjiä, on käyttöoikeuksien hallinta vaikeaa Excel-taulukkojen ja paperimappien kanssa. Tätä varten on tehty käyttöoikeuksien hallintajärjestelmiä. Käyttöoikeuksien hallintajärjestelmistä käytetään yleisesti myös nimityksiä käyttövaltuuksien hallintajärjestelmät, identiteettien hallintajärjestelmät (Identity Management, IDM) tai identiteettien ja pääsynhallinnanjärjestelmät (Identity and Access Management, IAM). (Propentus 2009, 4-6; VM 2006, 9-10; VM 2008, 161.)

Järjestelmien avulla tapahtuva käyttöoikeuksien hallinta mahdollistaa vakioidun toimintatavan. Isoissa organisaatioissa satojen tai tuhansien käyttöoikeuksien hallinta on järjestelmästä huolimatta vaikeaa ja oikeuksien ylläpito vaatii paljon työtä. Hallinnan helpottamiseksi on kehitetty roolipohjainen tapa hallita käyttöoikeuksia, jossa oikeudet haetaan aina roolin kautta. Roolipohjaisuus käyttöoikeuksien hallintajärjestelmissä tuo keinon hallita käyttöoikeuksia nopeammin, joustavammin ja tietoturvallisemmin. Roolipohjaisuutta tarkastellaan kappaleessa 3.

Käyttöoikeuksien hallintajärjestelmän avulla pyritään vastaamaan mm. seuraaviin kysymyksiin:

- Kuka hyväksyy käyttöoikeudet?
- Kuka vastaa käyttöoikeuksista?
- Pitääkö käyttöoikeus hyväksyttää useammalla taholla?
- Kenen pitää saada tieto myönnetystä käyttöoikeudesta?
- Kuka saa poistaa käyttöoikeuden?
- Kun henkilö vaihtaa tehtäviä, miten hallitaan muutostilanne?
- Mistä tiedetään henkilön työsuhteen ja käyttöoikeuksien alkaneen?
- Kuinka henkilö saa käyttöoikeudet aloittaessaan työtehtävät?
- Kun henkilö lähtee yrityksestä, kenen tehtäviin kuuluu oikeuksien poistot?
- Mistä tiedetään se, että henkilö on lähtenyt yrityksestä?
- Miten hallitaan pätkätyöläisten käyttöoikeudet?
- Kuinka hallitaan yrityksen ulkopuolisten henkilöiden käyttöoikeudet?

(Propentus 2009, 4-6.)

2.2 Yleinen arkkitehtuuri

Käyttöoikeuksien hallintajärjestelmien rakenne voidaan jakaa neljään pääkomponenttiin:

- keskitetty käyttäjä- ja käyttöoikeustietovarasto
- käyttöoikeuksien hyväksyntäprosessin hallinta
- automaattinen käyttövaltuustietojen provisiointi
- jäljitettävyys- ja raportointitoiminnot

Hallintajärjestelmän käyttäjätiedot ja käyttöoikeustiedot tallennetaan omaan keskitettyyn tietovarastoon, joka voi olla esimerkiksi tietokanta tai LDAP-hakemisto. Käyttöoikeuksien hallintajärjestelmien henkilötiedot perustuvat usein yhdestä tai useammasta ulkoisesta HR-järjestelmästä tai muusta henkilötietojärjestelmästä synkronoituihin tietoihin. (VM 2006, 24-26.)

Järjestelmien käyttöoikeuksia voidaan hakea automaattisesti perustuen esimerkiksi HR-järjestelmän tietoihin tai itsepalvelujärjestelmistä tapahtuviin loppukäyttäjien tai valtuutettujen käyttöoikeushenkilöiden pyyntöihin. Järjestelmään määritetään työnkulut, jotka sisältävät käyttöoikeuksien hyväksyntäprosessit. Käyttöoikeudet voivat vaatia esimerkiksi esimiehen ja sovelluksen pääkäyttäjän hyväksynnät ennen käyttöoikeuden myöntämistä. Hyväksymisen jälkeen käyttöoikeudet asetetaan kohdejärjestelmiin joko manuaalisesti tai automaattisesti. Manuaalinen tapa perustuu esimerkiksi pääkäyttäjille lähetettäviin sähköposteihin, joissa kerrotaan, mitä oikeuksia ja kenelle oikeudet tulee asettaa. Pääkäyttäjät kuittaavat tekemänsä toimenpiteet takaisin hallintajärjestelmään ja käyttäjä saa tiedon myönnetyistä käyttöoikeuksista. (VM 2006, 24-26.)

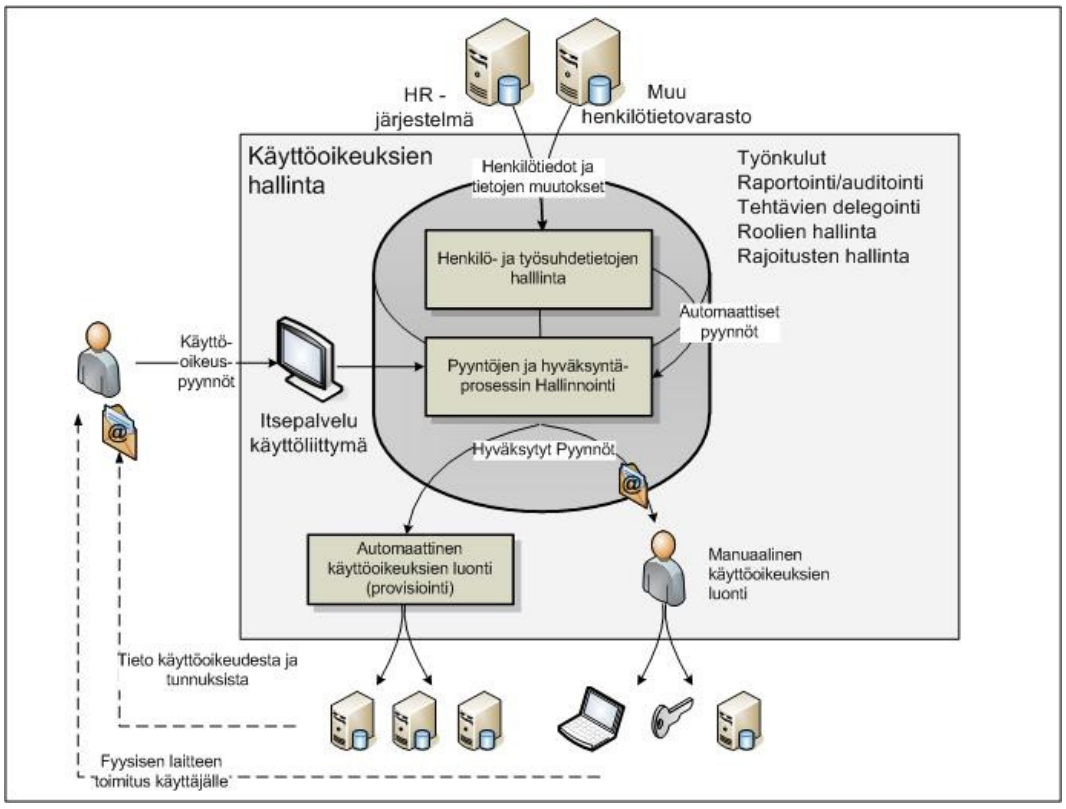
Vaihtoehtoinen tapa asettaa käyttöoikeudet kohdejärjestelmään on automaattinen provisiointi. Tällöin käyttöoikeuksien hallintajärjestelmän ja kohdejärjestelmän välille on luotu provisiointirajapinta, joka mahdollistaa automaattisen käyttöoikeuksien asettamisen. Rajapintojen toteuttamiseksi on myös kehitetty kansainvälisiä standardeja, kuten esimerkiksi SPML (Service Provisioning Markup Language). (VM 2006, 24-26.)

Järjestelmän yksi tärkeimmistä ominaisuuksista on raportointi. Kaikkia käyttöoikeuksien hallintajärjestelmien piirissä olevia tietoja ja tapahtumia tulee pystyä raportoimaan. Käyttöoikeuksien myöntämisestä, poistamisesta ja muutoksista tulee jäädä lokitiedot, joista voidaan selvittää toimenpiteiden tekijät ja hyväksyjät. Kattavan raportoinnin avulla on mahdollista seurata yksittäisten peruskäyttäjien ja erikoislaajoja oikeuksia omaavien käyttäjien käyttöoikeuksia. Järjestelmien tulisi myös pystyä varoittamaan käyttöoikeuksista, jotka eivät ole aktiivisia ja tarkastamaan käyttöoikeusmääritysten oikeellisuutta ja ajantasaisuutta. Ohje *Käyttövaltuushallinnon periaatteet ja hyvät käytännöt* (VM 2006, 26) tiivistää vaatimukset seuraavasti:

> Hallintajärjestelmän kautta tulee olla mahdollista saada milloin tahansa ajantasainen raportti toimintaympäristössä käytössä olevista käyttäjäidentiteeteistä, niiden haltijoista ja niihin liittyvistä käyttövaltuuksista samoin kuin palvelujärjestelmissä määritellyistä suojattavista kohteista ja niihin liittyvistä käyttövaltuuksista.

Pääkomponenttien lisäksi käyttöoikeuksien hallintajärjestelmissä on ominaisuuksia, joiden avulla voidaan hallita mm. käyttöoikeuksien hyväksyntätehtävien delegointia toisille henkilöille, roolien käsittelyä ja erilaisten rajoitusten, kuten esimerkiksi kiellettyjen yhdistelmien hallintaa.

Kuviossa 1 on selvennetty arkkitehtuurin ja työnkulun periaatetta. Järjestelmässä käytetyt henkilötiedot saadaan HR-järjestelmästä tai ulkoisesta tietojärjestelmästä. Käyttäjät tekevät käyttöoikeuspyyntöjä itsepalvelukäyttöliittymästä tai pyyntöjä voi generoitua automaattisesti myös HR-tietojen perusteella. Pyynnöt ohjataan hyväksyntään, minkä jälkeen käyttöoikeuksia luodaan kohdejärjestelmiin joko täysin automaattisesti tai oikeuksien asettaminen vaatii manuaalisia toimenpiteitä esimerkiksi fyysisten laitteiden osalta. Uusista käyttöoikeuksista lähetetään palaute käyttäjälle esimerkiksi sähköpostitse.



KUVIO 1. Käyttöoikeuksien hallintajärjestelmän yleinen arkkitehtuuri ja toimintaperiaate

Yllä esitetty malli soveltuu sekä työvuoperiaatteella (eng. workflow) toimivien että enemmän automatiikka sisältävien järjestelmien periaatteeksi. Työvuoperiaatteella toimivissa järjestelmissä käyttöoikeuksista tehdään pyyntö, joka joudutaan hyväksyttämään ennen käyttöoikeuden antamista. Hyväksyntäketjuun voi kuulua esimerkiksi henkilön esimies ja järjestelmän vastuuhenkilö. Muutostilanteissa käyttöoikeuksista tehdään uusia pyyntöjä.

Työnkulkua voidaan automatisoida myöntämällä käyttöoikeuksia ja rooleja automaattisesti perustuen henkilötietoihin esimerkiksi titteliin tai organisaatioon. Automaattisessa työnkulussa hyväksyntäketju on minimoitu ja käyttöoikeudet luodaan kohdejärjestelmiin automaattisen provisioinnin avulla. Myös muutostilanteissa muutokset voidaan viedä kohdejärjestelmiin automaattisesti.

2.3 Käyttöoikeuksien hallinnan vaikeudet ja ongelmat

Käyttöoikeuksien hallinnan ongelmia kuvataan valtiovarainministerin ohjeessa *Käyttövaltuushallinnon periaatteet ja hyvät käytännöt* vuodelta 2006. Perinteisessä käyttöoikeuksien hallinnassa prosesseja ja vastuita ei ole kuvattu tarkasti. Järjestelmä- ja prosessivastuu on usein delegoitu täysin tietohallinnolle. Käyttöoikeuksien myöntämisen prosessia ei kontrolloida kunnolla, ja henkilöille annetaan varmuuden vuoksi liian suuret oikeudet. Huonon kontrollin vuoksi työtehtävistä poistuneiden henkilöiden oikeudet voivat jäädä voimaan jopa vuosiksi. Tietoturvariskit ja mahdollisuus väärinkäytöksiin kasvaa. Käyttöoikeuksien hallinta on yleensä hyvin manuaalista työtä ja järjestelmävastaavat tekevät sen. Dokumentointi käyttöoikeustapahtumista ja muutoksista ei ole riittävää ja jäljitettävyys vaikeaa. (VM 2006, 9-10.)

Isoissa organisaatioissa uuden henkilön käyttöoikeuspyynnön suorittamiseen tai olemassa olevien käyttöoikeuksien muuttamiseen voi liittyä usean henkilön muodostama ketju. Jokaisella järjestelmällä voi olla eri järjestelmänvalvojat ja eri järjestelmään kohdistuvat toimet vaativat erilliset työpyynnöt. Mitä enemmän mukana on eri osapuolia, sitä kauemmin käyttöoikeuspyynnöt kestävät, ja pahimmassa tapauksessa yksittäisen käyttöoikeuspyynnön läpimeno voi kestää viikkoja. Nykyään yhä enenevissä määrin yritysten ulkopuolisille käyttäjille tulee pystyä antamaan käyttöoikeuksia. Ulkopuolisten toimijoiden kanssa tehokkuus- ja aikavaatimukset ja työn dynaaminen luonne eivät salli perinteisen jäykkää käyttöoikeuksien hallintaa. Yritysten pitää myös pystyä vastaamaan yhä useammin tapahtuvien uudelleen organisoitumisten aiheuttamiin muutoksiin käyttöoikeuksissa. (Mienes 2003, 2.)

Perinteisissä käyttöoikeuksien hallinnan malleissa jokaisella järjestelmällä on omat omistajat ja pääkäyttäjät, joiden vastuulla on hallita käyttöoikeuksia. Käyttöoikeuksia ylläpidetään esimerkiksi käyttäjäryhmien tai käyttäjäprofiilien avulla, jotka määrittävät, mitä näyttöjä, toimintoja tai valikoita henkilö voi kohdejärjestelmässä käyttää. Pyynnöt käyttöoikeuksista tehdään usein esimiesten toimesta samoin kuin poistopyynnötkin. Tällaisen ympäristön käyttöoikeuksien hallinta

perustuu järjestelmien omistajien, pääkäyttäjien ja esimiesten manuaaliseen työhön. (Mienes 2003, 2.)

Käyttöoikeuksien pyytäjän kannalta tilanne on myös haastava, koska usein joudutaan pyytämään oikeuksia kymmeniin eri järjestelmiin. Tällöin käytetään usein periaatetta, jossa henkilölle myönnetään samat oikeudet kuin hänen työtoverillaan on, eikä tutkita tarkemmin, onko kaikki oikeudet tarpeellisia. Toiminta on tehokasta, mutta henkilöt saavat usein liian suuret oikeudet. Ongelmana on myös usein käyttöoikeustilanteiden tarkastaminen. Jokaisen järjestelmän roolit ja oikeudet on muodostettu ja asetettu eri tavalla ja eivätkä ne ole usein ajan tasalla. Tällöin käyttöoikeuksien tarkastaminen automaattisesti on mahdotonta. (Mienes 2003, 2-3.)

Käyttöoikeuksien hallinnassa on myös ongelmana kenelle vastuu käyttöoikeuksien hallinnasta kuuluu. Perinteisesti yritysten tietohallinto vastaa käyttöoikeuksien hallinnasta, koska tietohallinto ylläpitää järjestelmiä ja usein vastaa myös järjestelmien käyttöoikeuksien, käyttäjäryhmien ja roolien muodostamisesta. Käyttöoikeuksien myöntäminen on taas esimiesten tehtävä, jotka parhaiten tietävät, mitä tehtäviä alaiset työssään suorittavat ja mitä käyttöoikeuksia tai rooleja he tarvitsevat. Tietohallinto ei välttämättä usein edes tiedä, onko heille tullut käyttöoikeuspyyntö aiheellinen vai ei. On vain luotettava pyynnön tekijään. Käyttöoikeuksien hallintajärjestelmät perustuvat usein yritysten henkilöstötietojärjestelmistä saataviin tietoihin ja näiden perusteella voidaan tarvittaessa luoda automaattisia käyttöoikeuksia. Tässä tapauksessa vastuu käyttöoikeuksista siirtyy henkilöstöosastolle. Yrityksien tapa hallita käyttöoikeuksia määrittää, kuka todellisuudessa vastaa käyttöoikeuksista.

Puutteellisesta käyttöoikeuksien hallinnasta seuraa useita ongelmia:

- Käyttäjillä on liian laajat käyttöoikeudet
- Eri järjestelmissä on erilaiset käyttöoikeuksien myöntämisen prosessit
- Käyttöoikeuksien pyytäminen ja hyväksyminen on epäselvää
- Käyttöoikeuksien hallittavuus on vaikeaa
- Käyttöoikeuksien tarkastaminen ja raportointi on vaikeaa

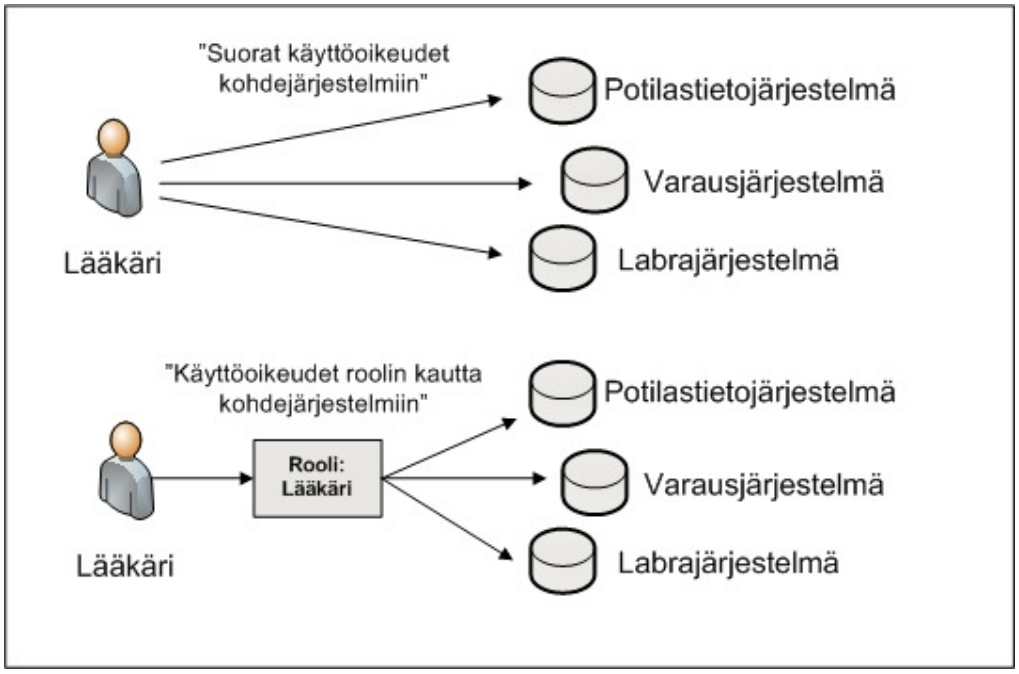
- Käyttöoikeuksien saaminen kestää
- Käyttöoikeuksien päättyminen ajoissa ja luotettavasti ei onnistu

Tämä kaikki johtaa heikentyneeseen tehokkuuteen, suurentuneisiin kustannuksiin ja haavoittuvuuteen tietoturvallisuudessa. (Mienes 2003, 2-3; Hitachi 2010.)

3 ROOLIPOHJAINEN KÄYTTÖOIKEUKSIEN HALLINTA

3.1 Perusperiaate

Ferraiolo ja kumppanit esittivät vuonna 1992 roolien perusajatuksen: Rooli on joukko toimenpiteitä, joita henkilöt voivat suorittaa organisaatiossaan. Roolilla tarkoitetaan organisaatioon kuuluvaa työnkuvaa tai titteliä, jossa määritetään roolien mukaisissa tehtävissä toimivien henkilöiden valtuudet ja vastuut. Yksinkertaistettuna rooli on joukko käyttöoikeuksia, joita henkilö tarvitsee suoriutuakseen työtehtävistään. Verrattuna käyttöoikeuksien hallintaan ilman rooleja on roolien periaate hyvin yksinkertainen. Käyttöoikeudet asetetaan kuuluvaksi rooleihin mieluummin kuin suoraan käyttäjille ja käyttäjille asetetaan mieluummin rooleja kuin suoria käyttöoikeuksia. Periaatetta on selvennetty kuviossa 2. (Ferraiolo & Kuhn 1992, 4; Sandhu, Coynek, Feinsteink & Youmank 1996, 1.)



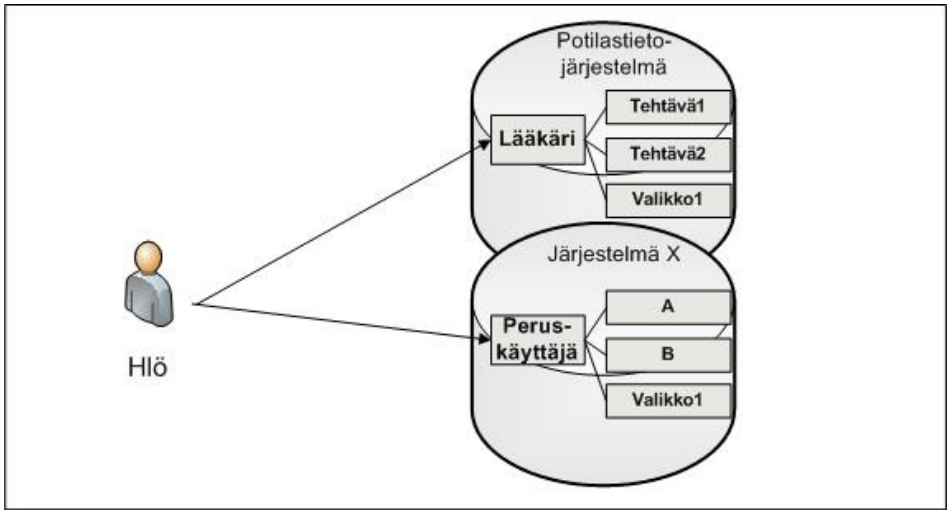
KUVIO 2. Suorien ja roolin kautta myönnettävien käyttöoikeuksien periaatteet

3.2 Työroolit ja järjestelmäroolit

Roolien perusperiaate on edellä mainitun yksinkertainen, mutta roolien käyttötarkoitus ei ole yhtä selkeä. Roolit voidaan jakaa karkeasti kahteen ryhmään käyttötarkoituksensa perusteella:

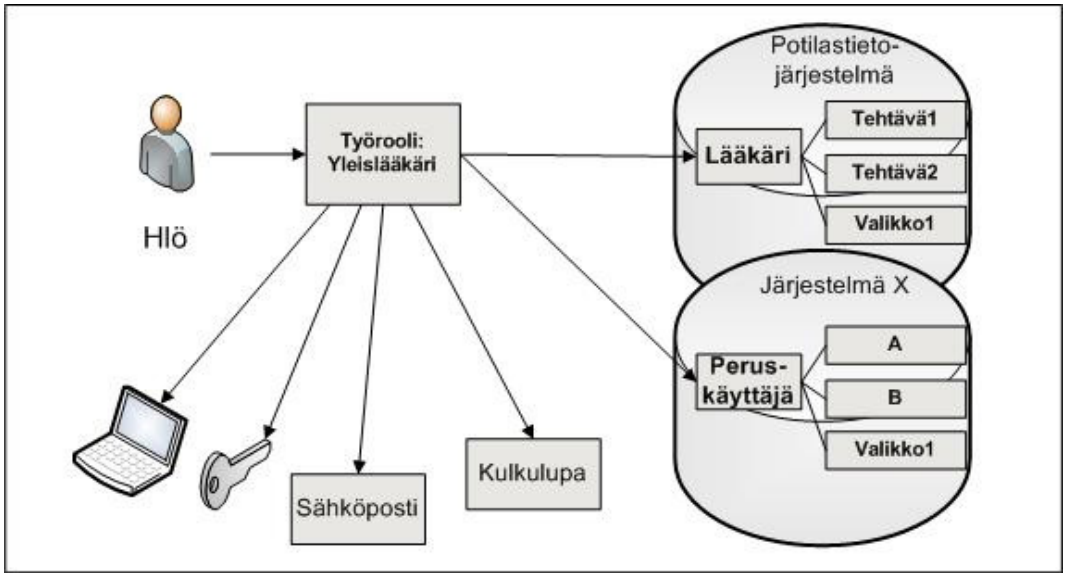
- järjestelmäroolit; järjestelmien sisäisten käyttöoikeuksien hallinta roolien avulla
- työroolit; organisaatioiden kaikkien käyttöoikeuksien hallinta roolien avulla

Järjestelmien sisäisten oikeuksien hallinnan rooleja kutsutaan usein *järjestelmärooleiks*i ja organisaatiotason rooleja *työrooleiksi*. Näillä rooleilla on merkittävä ero käyttötarkoituksessa. Järjestelmärooleilla kuvataan, mitä toimenpiteitä yksittäisessä kohdejärjestelmässä voi tehdä. Roolin sisältämiä oikeuksia voi olla esimerkiksi lukuoikeudet tiettyyn ohjelmaan, tietyt valikko-oikeudet tai oikeus vahvistaa tilauksia. Periaatteeltaan tämä ei eroa perinteisistä käyttäjäryhmistä, joiden avulla usein määritellään järjestelmissä käyttäjien käyttöoikeudet. Kuviossa 3 selkeytetään järjestelmäroolien käyttötarkoitusta. Järjestelmärooli on tarkoitettu yhden järjestelmän sisäiseen käyttöoikeuksien hallintaan.



KUVIO 3. Järjestelmäroolien periaate

Työrooleilla tarkoitetaan yleensä henkilöiden työtehtäviin, organisaatioyksikköön, asemaan tai titteliin liittyviä rooleja, joilla on tarkoitus kuvata, mitä käyttöoikeuksia henkilö tarvitsee suoriutuakseen työtehtävistään. Työroolin sisältämät käyttöoikeudet voivat olla toisia rooleja, tietojärjestelmien järjestelmärooleja, käyttäjäryhmiä tai rooleihin liitettyjä suoria käyttöoikeuksia järjestelmiin, sovelluksiin tai fyysisiin käyttöoikeuksiin, kuten esimerkiksi kulkulupiin. Työroolit eivät ota kantaa, mitä yksittäisiä operaatioita kohdejärjestelmissä voi tehdä. Kuviossa 4 on esitetty työroolien periaate. Työroolien roolien avulla on tarkoitus hallita yritysten tai organisaatioiden kaikkia käyttöoikeuksia. Työroolista voidaan käyttää myös termiä *koosterooli* (Propentus 2009).



KUVIO 4. Työroolien periaate

Roolipohjaisessa käyttöoikeuksien hallinnassa tehokkain ja joustavin keino on käyttää työrooleja sekä järjestelmärooleja yhdessä. Roolien määrittely joudutaan tekemään kummallekin erikseen, ja mikä vaatii suuren kertaluontoisen työn. Tämän jälkeen hallinnollinen työ keskittyy käyttöoikeuksien hallinnoijien kannalta työroolien ja järjestelmäroolien väliseen kytkemiseen ja suoraan järjestelmiin kohdistuva ylläpitotarve vähenee. (VM 2006, 17-18.)

Termiä RBAC (Role-based Access Control) käytetään usein yleisenä terminä puhuttaessa rooleista riippumatta roolien käyttötarkoituksesta. RBAC:n perusperiaate sopii sekä työ että järjestelmäroolien käyttöön, mutta RBAC-standardit ja -mallit ovat usein tehty yksittäisen järjestelmän pääsynhallintaa varten. RBAC-standardeihin on sittemmin esitetty useita eri laajennuksia, kun on huomattu tarpeelliseksi hallita koko organisaation kaikkia rooleja, eikä keskittyä yksittäisten järjestelmien roolien hallintaan. RBAC-standardeja ja -laajennuksia käsitellään tarkemmin kappaleessa 5.

3.3 Tavoitteet

3.3.1 Hallittavuus ja ylläpito

Roolien mukaisella käyttöoikeuksien hallinnalla tavoitellaan ensisijaisesti parempaa hallittavuutta. Perinteinen malli, jossa käyttöoikeudet asetetaan henkilöille suoraan, on sitä vaikeampi hallita, mitä laajemmista tietojärjestelmäkokonaisuuksista ja organisaatioista on kyse. Roolien periaatteena on antaa käyttöoikeudet henkilöille roolien kautta. Tutkimukset ovat osoittaneet sen, että henkilöt ja henkilöihin liittyvät roolit muuttuvat useammin kuin itse roolien sisältö. Tutkimuksen mukaan on myös järkevämpää antaa järjestelmänhaltijoiden asettaa henkilöille esimääritettyjä rooleja kuin luoda uusia rooleja tai roolien sisältämiä oikeuksia. Roolien asettaminen henkilöille ei vaadi myöskään yhtä paljon teknistä osaamista, kuin yksittäisten käyttöoikeuksien luominen kohdejärjestelmiin. Roeckle ja kumppanit toteavatkin sen, että käyttöoikeuksien hallitsemiseksi hyvin tulee oikeuksien asettamisen ja poistamisen prosessin olla helppo, hyvin kuvattu ja nopea. (Sandhu ym. 1996,1; HL7 2005, 5; Roeckle, Shimpf & Weidinger 2000, 2.)

Paremman hallittavuuden lisäksi käyttöoikeuksien myöntäminen roolien kautta on myös nopeampaa kuin käyttöoikeuksien asettaminen yksitellen. Tämä nopeuttaa uusien työntekijöiden käyttöoikeuksien saamista ja toisaalta käyttöoikeuksien poistumista työsuhteen päättyessä. (Sandhu 1996, 1; Gonzales-Webb 2007, 14.)

3.3.2 Tietoturva

Roolien avulla voidaan määrittää rajoituksia henkilön käyttöoikeuksille. Se mitä henkilö pystyy työtehtävässään tekemään, ei ole sama, kuin mitä henkilön kuuluu ja mitä henkilö saa tehdä. Roolien avulla voidaan toteuttaa tärkeää käyttöoikeuksien hallinnassa käytettyä vähäisimpien oikeuksien periaatetta (eng. least priviledges) tai muita rajoituksia, kuten esimerkiksi estää käyttöoikeuksien kiellettyjä yhdistelmiä. Käyttöoikeuksien hallintaa roolien kautta pidetäänkin tietoturvallisempana vaihtoehtona kuin yksittäisten käyttöoikeuksien avulla tapahtuvaa hallintaa. (Sandhu ym. 1996, 1; HL7 2005, 5.)

Roolit antavat myös paremman mahdollisuuden raportoida käyttäjien käyttöoikeuksia sekä helpottavat auditointia, jossa joudutaan tarkastamaan onko henkilöille myönnetyt käyttöoikeudet sallittuja. Kattava raportointi ja säännöllinen auditointi ennaltaehkäisevät mahdollisia tieturvauhkia. Roolien avulla tapahtuva käyttöoikeuksien hallinta vähentää auditoitavien tapahtumien määrää. (Mienes 2003, 9; HL7 2005, 5; Kern 2002, 4; Sandhu ym. 1996, 1.)

3.3.3 Vastuun siirtäminen oikeaan paikkaan

Roolit antavat mahdollisuuden jakaa vastuuta tietohallinnon ja liiketoimintayksiköiden välillä. Tietohallinto omaa tietotaidon luoda rooleja ja roolien sisältämiä oikeuksia, mutta ei välttämättä tiedä, mitä rooleja kukin henkilö tarvitsee. Roolien asettaminen henkilöille kuuluu taas esimiehille, jotka tietävät paremmin henkilöiden työtehtävät ja niiden vaatimat roolit. (Kern 2002, 8.)

3.3.4 Kustannukset

Hallinnollisten kustannusten väheneminen on suoraan verrannollinen tietohallinnon työmäärään käyttöoikeuksien hallinnassa. Roolien myötä toimenpiteet kohdis-

tuvat roolien asettamiseen käyttäjille ja vähentävät suoria järjestelmiin kohdistuvia toimenpiteitä. Roolipohjaisuus antaa mahdollisuuden automatisoida käyttöoikeuksien hallintaa. Rooleja voidaan myöntää tai poistaa automaattisesti, ja järjestelmien väliset toimenpiteet voidaan myös automatisoida. Automatisoinnin aste on riippuvainen organisaation tietoturvapolitiikasta, tietojärjestelmien yhteensopivuudesta ja käyttöoikeuksien hallinnan prosessista. Työvuoperiaatteella toimiva käyttöoikeuksien hallinta vaatii joka tapauksessa manuaalista työtä käyttöoikeuksien hyväksyntäprosessin osalta. Pitkälle vietynä automatisoinnilla on suuri merkitys. Eräässä tapauksessa kooltaan 40 000 työntekijän suuruinen Eurooppalainen pankki oli onnistunut automatisoimaan käyttöoikeuksien hallinnan työstä 95%. (Gonzales-Webb 2007, 14; Kern 2002, 4-8.)

Yrityksissä on yleistä käyttöoikeuksien kertyminen työtehtävien ja organisaatioiden vaihtuessa. Poistuvien käyttöoikeuksien hallitseminen helpottuu roolien käyttöönoton seurauksena. Henkilöltä voidaan poistaa rooli ja sitä kautta kaikki roolin sisältämät käyttöoikeudet nopeasti. Puustinen (2008) on tutkimuksessaan viitannut siihen että 30-60% käyttöoikeuksista ei ole päteviä tai ajan tasalla. Tämä tarkoittaa suoraan sitä, että tarkalla ja ajantasaisella käyttöoikeuksien hallinnalla voidaan vähentää merkittävästä hallinnollista työtä sekä ylimääräisiä lisenssikustannuksia johtuen käyttäjillä olevista turhista lisensseistä. (Puustinen 2008, 32.)

4 SÄÄDÖKSET

4.1 Yleistä

Jokaisen yrityksen ja organisaation tulee huolehtia järjestelmien käyttöoikeuksien hallinnoinnista ja määrittää käyttövaltuushallinnan periaatteet. Tämän lisäksi Suomessa on olemassa useita yritysten tietoturvaan ja käyttöoikeuksien hallintaan vaikuttavia erilaisia lakeja ja säädöksiä. Tiedonhallintaa koskeva Julkisuuslaki (621/1999) ja henkilötietojen käsittelyä säätelevä henkilötietolaki (523/1999) asettavat säädöksiä henkilötietojen suojaamiselle, tarpeellisuus- ja virheettömyysvaatimukselle sekä käyttötarkoitussidonnaisuudelle. Lait edellyttävät yrityksien järjestelmiltä asianmukaista käyttöoikeuksien hallintaa ja valvontaa sekä käyttöoikeuksien käsittelemistä henkilötasolla. (VM 2006, 11-12.)

Käyttöoikeuksien hallintaan vaikuttavat myös perustuslaki, laki sosiaali- ja terveyden huollon asiakastietojen sähköisestä käsittelystä, laki sähköisestä lääkemääräyksestä, julkisuuslaki sekä monet terveydenhuollon erityislait. Tutkimuksessa asiakastapauksena olevan Salon kaupungin käyttöoikeuksien hallinnan piiriin kuulu olennaisena osana terveydenhuollon henkilöstö, joten tässä tutkimuksessa keskitytään terveydenhuoltoon liittyviä lakeihin ja säädöksiin ja niiden vaikutukseen roolipohjaiseen käyttöoikeuksien hallintaan. Kappaleessa 4.3 listataan yhteenveto, mitä vaatimuksia Suomen lait asettavat potilastiedon ja terveyteen liittyvän henkilötiedon sähköiselle käsittelylle.

Käyttöoikeuksien hallintajärjestelmien kehitykseen ja roolipohjaiseen käyttöoikeuksien hallintaan vaikuttavista ulkomaisista säädöksistä tärkein on kappaleessa 4.2 kuvattu laki *Sarbanes-Oxley Act 2002* (SOX). Muita käyttöoikeuksien hallintaan vaikuttavia ulkomaisia säännöstöjä ovat mm. Yhdysvaltain potilastietojen tietoturvan säännöstö *Health Insurance Portability and Accountability Act (HIPAA),* joka suosittelee yhdeksi pääsynhallinnan malliksi RBAC-mallia. Pankkialan sään-

nöstönä toimii *Gramm-Leach-Bliley Act (GLB).* Näitä lakeja ei kuitenkaan kuvata tässä tutkimuksessa tarkemmin. (HIPAA 2010, GLB 2010.)

4.2 Sarbanes-Oxley Act (SOX)

Vuonna 2002 Yhdysvalloissa voimaan tullut laki Sarbanes-Oxley Act (SOX) sisälsi joukon tiukkoja vaatimuksia, joiden avulla pyrittiin estämään yrityksien tietojärjestelmien avulla tapahtuvia taloudellisia petoksia. 2000-luvun alkupuolella tapahtui suuria yrityspetoksia (mm. Enron, Tyco International, Adelphia, Peregrine Systems ja WorldCom), joiden seurauksena SOX-laki kehitettiin. Petokset johtuivat lähinnä yritysten sisäisen valvonnan heikkouksista. Lain seurauksena yritysten tulee tietää, kuka heidän tietojärjestelmiään käyttää, milloin järjestelmiin on kirjauduttu sisään ja ulos, mitä järjestelmissä on tehty, mitä muutoksia on tapahtunut ja millä valtuuksilla järjestelmiä on käytetty. Näiden vaatimuksien täyttämiseksi yritykset ovat panostaneet käyttöoikeuksien hallinnan parantamiseen käyttöoikeuksien hallintajärjestelmien avulla. (RBAC & Sarbanes-Oxley Compliance; Bednarz 2005; KPMG 2004.)

Käyttöoikeuksien hallinnan helpottamiseksi tehty roolipohjainen pääsynhallinnan standardi Role-Based Access Control (RBAC) on suunniteltu erityisesti ratkaisemaan SOX-lain vaatimukset. SOX-vaatimukset täyttävistä tietojärjestelmistä pystytään jälkikäteen auditoimaan ja raportoimaan käyttäjien tekemät järjestelmätapahtumat ja määrittämään, onko henkilöillä ollut riittävät valtuudet tapahtumien suorittamiseen. SOX-vaatimukset yrityksien käyttöoikeuksien hallintaan voidaan tiivistää seuraavasti:

- Yrityksillä tulee olla kattava prosessi tietojärjestelmien käyttöoikeuksien valvontaan sekä hallintaan, ja käyttöoikeuksien tarkastaminen täytyy suorittaa säännöllisesti.
- Yrityksillä tulee olla tehokas ja turvallinen prosessi henkilöiden käyttöoikeuksien myöntämiseen ja poistamiseen.
- Yritysten on pystyttävä varmistamaan sen, että arkaluontoiseen tietoon on pääsy vain henkilöillä, joilla on käyttöoikeudet.

(Bednarz 2005; KPMG 2004.)

SOX laki sisältää 11-kohtaisen säännöstön mm. yritysten johdon vastuista ja velvollisuuksista sekä lain rikkomisesta seuraavista rangaistuksista. Erityisesti lain kohta 404 velvoittaa yhtiöiden johtoa luomaan ja ylläpitämään tehokasta sisäistä valvontaa, joka vaikuttaa suoraan tietojärjestelmien käyttöoikeuksien hallinnan tehostamiseen. (Bednarz 2005; KPMG 2004.)

SOX-laki koskee kaikkia Yhdysvaltalaisia ja ulkomaalaisia yrityksiä, jotka on listattu Yhdysvaltain arvopaperimarkkinoita valvovan elimen SEC:in (The Securities and Exchange Commission) alaisessa pörssissä (KPMG 2004). Vastaavanlaisia lakeja ja säännöksiä on myös laadittu mm. Kanadassa, Saksassa, Ranskassa ja Japanissa. (RBAC & Sarbanes-Oxley Compliance; Bednarz 2005; KPMG 2004.)

4.3 Potilastietojen sähköisen käsittelyn säädökset

Käyttöoikeuksien hallinta terveydenhuollon toimialalla sisältää usein potilastietoihin kohdistuvia toimintoja. Stakesin raportissa Ruotsalainen (2006) tiivistää, mitä vaatimuksia Suomen lainsäädäntö asettaa potilasasiakirjatiedon ja muun terveyteen liittyvän henkilötiedon sähköiselle käsittelylle:

- Tietojen käyttöön, talletukseen, ylläpitoon ja luovutukseen tulee olla suunnitelmallista.
- Tiedot eivät saa joutua ilman suostumusta tai laista johtuvaa muuta perustetta sivullisten käsiin.
- Terveydenhuollon ammattihenkilö tulee voida tunnistaa ja todentaa sähköisessä asioinnissa.
- Potilas/asiakas tulee voida tunnistaa ja todentaa sähköisessä asioinnissa
- Toimintayksiköt, palvelimet ja muut entiteetit tulee voida tunnistaa ja varmentaa.
- Vain hoidon kannalta tarpeellisia tietoja saa käsitellä.

- STM:n (Sosiaali- ja Terveysministeriö) normiannolla erikseen määritellyissä potilasasiakirjoissa tulee olla omakätinen tai sähköinen allekirjoitus.
- Tietojen käsittelyn edellytys on hoitosuhde, asiayhteys tai muu laista johtuva peruste.
- Jollei laista muuta johdu, ei henkilötietoa saa käyttää muuhun käyttötarkoitukseen, kuin mihin ne on kerätty.
- Potilaalla on oikeus määrätä (lain säätämissä rajoissa) omien terveystietojen käytöstä ja luovutuksesta.
- Rekisterinpitäjän tulee voida seurata tietojen käyttöä ja luovutusta.

(Ruotsalainen 2006, 53.)

4.4 Kansallisen Terveysarkiston (KanTa) vaatimukset

Suomessa säädettiin vuonna 2007 Laki sosiaali- ja terveydenhuollon sähköisestä käsittelystä sekä Laki sähköisestä lääkemääräyksestä. Lakien avulla pyrittiin edistämään sosiaali- ja terveydenhuollon asiakastietojen tietoturvallista sähköistä käsittelyä ja arkistointia sekä potilas- ja lääketurvallisuutta. Lakien seurauksena Suomessa alettiin kehittää uusia sähköisiä palveluja ja lopputuloksena syntyi Kansallinen Terveysarkisto *KanTa*. Hanke oli Sosiaali- ja terveysministeriön koordinoima ja mukana kehittämässä olivat myös kansaneläkelaitos Kela, kuntien tiedotuskeskuksen organisaatio KunTo-toimisto, Sosiaali- ja terveysalan lupa- ja valvontavirasto Valvira, terveyden ja hyvinvoinnin laitos THL sekä eri terveydenhuollon toimijoita ja tietojärjestelmätoimittajia. KanTa-palvelun sisältönä ovat valtakunnalliset tietojärjestelmäpalvelut sähköinen resepti *eResepti*, kansallinen lääketietokanta, sähköinen potilastiedon arkisto *eArkisto* ja kansalaisten omien resepti- ja potilastietojen katselu *eKatselu*. Vuonna 2010 *eResepti* on pilotointivaiheessa ja *eArkisto* kehitysvaiheessa. (KanTa 2010.)

Edellä mainittujen lakien mukaan kaikilla julkisen terveydenhuollon palvelun antajilla on velvollisuus liittyä KanTa-palvelujen käyttäjiksi 1.4.2011 mennessä eReseptin ja eArkiston osalta. Sama koskee myös yksityisiä terveydenhuollon pal-

velujen antajia, jos he toteuttavat arkistoinnin pitkäaikaissäilytyksen sähköisesti. (Laki sosiaali- ja terveydenhuollon sähköisestä käsittelystä 159/2007, 25§, Laki sähköisestä lääkemääräyksestä 61/2007, 28§.)

Kanta-palvelun verkkosivuilla kuvataan eArkiston tarkoitus seuraavasti: ” Sähköinen potilastiedon arkisto (eArkisto) tulee tarjoamaan terveydenhuollon organisaatioille (esim. terveyskeskus, yksityinen lääkäriasema) keskitetyn sähköisten potilastietojen arkiston ja hoitotietojen saatavuuden yli organisaatiorajojen potilaan suostumuksella.” Palvelua käytetään apteekki- ja potilaskertomusjärjestelmien kautta. Tämä aiheuttaa kovia vaatimuksia kyseisten järjestelmien tietoturvalle ja käyttöoikeuksien valvonnalle, jotta varmistetaan arkaluontoisten potilastietojen lainmukainen käyttö. (KanTa 2010.)

Stakesin raportissa Ruotsalainen (2006) kuvaa vaatimuksia sähköistä arkistoa käyttäville toimintayksiköille: ”Kaikilla terveystietoja arkistoivalla toimintayksiköllä tulee olla hallinnollinen, tekninen ja fyysinen infrastruktuuri, jotta se voi varmistaa tiedon muuttumattomuuden, tietojen luottamuksellisuuden ja käytettävyyden sekä yksityisyyden suojan toteutumisen koko tietojen säilytysajan.” Raportin mukaan jokaisella arkistoa käyttävällä toimintayksiköllä tulee olla myös kirjallisesti määritetty tietoturvapolitiikka. Osana tätä tietoturvapolitiikkaa tulee käyttöoikeuksia hallita roolipohjaisesti. Sähköisellä arkistolla tulee olla tietojen luovutuksenhallintajärjestelmä sekä tiedon käytön valvontajärjestelmä, joiden periaatteisiin kuuluu arkiston tietoja luovuttavan tai käyttävän henkilön ja henkilön roolin tunnistaminen. (Ruotsalainen 2006, 26-29.)

Henkilöiden tunnistamista ja käyttöoikeuksia hallitaan yleensä jokaisen toimintayksikön sisäisillä tietojärjestelmillä. Toimintayksiköissä toimii myös usein toimintayksikön ulkopuolisia käyttäjiä, jotka myös toimivat eri järjestelmien loppukäyttäjinä. Myös näiden käyttäjien osalta on varmistettava henkilöiden tunnistaminen ja tietojen lainmukainen käyttäminen sekä valvonta (Ruotsalainen 2006, 52). Nämä vaatimukset korostuvat erityisesti käsiteltäessä arkaluontoisia tietoja, kuten potilastietoja, sähköstä arkistoa tai sähköisiä lääkemääräyksiä. Roolien mu-

kainen käyttöoikeuksien hallinta koskee siis myös organisaation ulkopuolisia käyttäjiä.

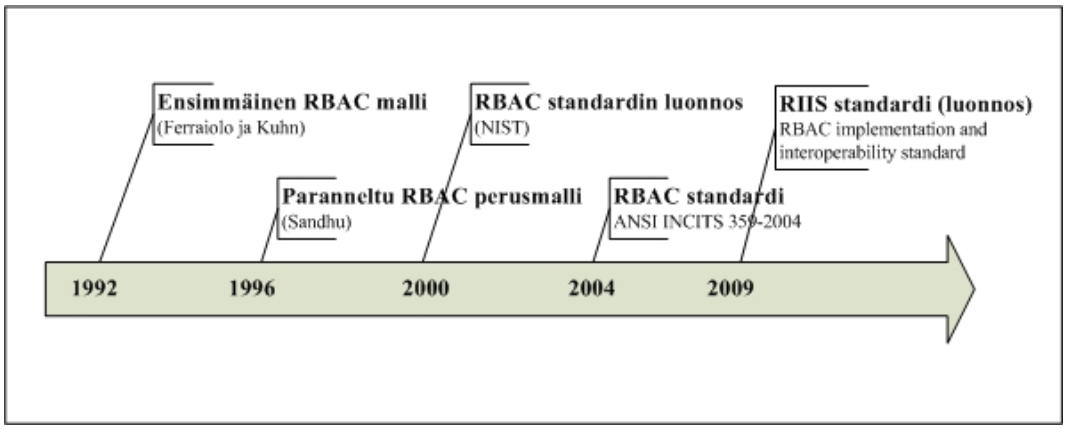
Laki sähköisestä lääkemääräyksestä (61/2007) ottaa myös kantaa tietotekniseen toteutukseen ja käyttöoikeuksien hallintaan. Lain luvun 20 § mukaan sähköinen lääkemääräys tulee toteuttaa siten, että ”reseptikeskuksessa olevien tietojen katselu, tallettaminen ja muu käsittely edellyttää käsittelijän yksilöivää vahvaa tunnistusmenetelmää sekä järjestelmään liittyvää käyttöoikeuksien hallintaa.”

5 RBAC-STANDARDIT JA MALLIT

5.1 Yleistä

Roolien hallinnassa käytetään usein termiä termiä RBAC (eng. Role-based Access Control). Termillä RBAC tarkoitetaan joko työroolien hallintaa, eli roolien käyttöoikeuksien hallintaa työtehtävien näkökulmasta huomioiden kaikki järjestelmät, tai termiä käytetään viitatessa yksittäisen järjestelmän pääsynhallintaan.

RBAC-standardeilla kuvataan jälkimmäistä, eli kuinka roolien mukainen käyttöoikeuksien hallinta tulisi huomioida yksittäisessä järjestelmässä. RBAC-standardien avulla on haluttu kehittää yhteinen tapa, jolla eri järjestelmien pääsynhallinta tulisi toteuttaa roolipohjaisesti. Yhtenäinen malli mahdollistaa samojen roolimääritysten käyttämisen eri järjestelmissä. Ensimmäinen RBAC-malli esitettiin vuonna 1992, ja standardin aseman RBAC saavutti 2004. Kuviossa 5 esitetään mallien kehitys aikajanalla. Mallit kuvataan tarkemmin seuraavissa kappaleissa.



KUVIO 5. Tärkeimmät RBAC-mallit

5.2 Ensimmäinen RBAC-malli 1992

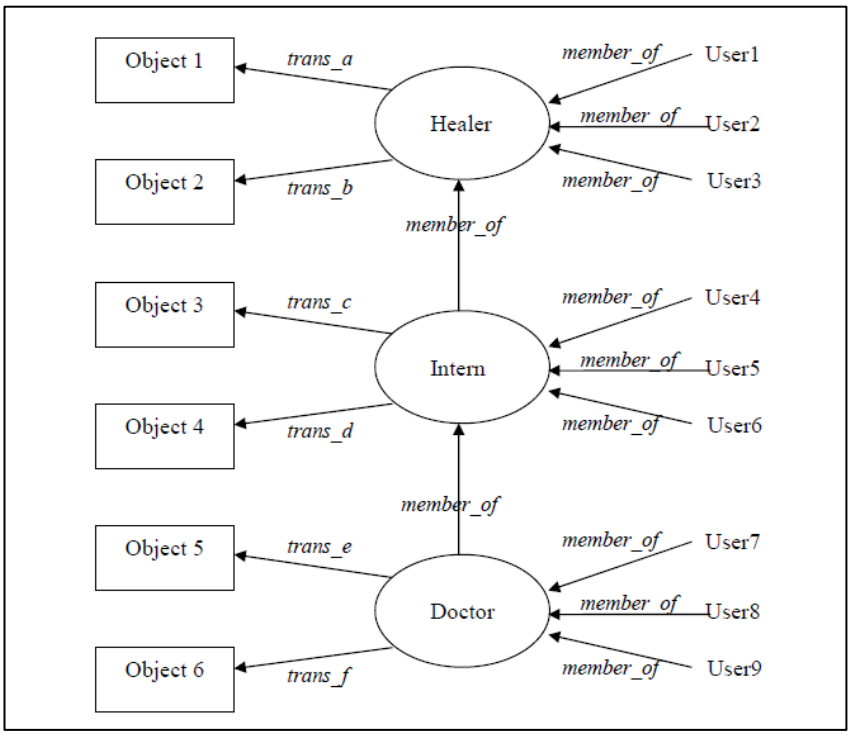
Ennen roolipohjaista käyttöoikeuksien hallintaa tietojärjestelmien pääsynhallinnan mallit voitiin jakaa kahteen luokkaan: Pakollinen pääsynhallinta MAC (eng. Mandatory Access Control) ja harkinnanvarainen pääsynhallinta DAC (eng. Discretionary Access Control). MAC oli sopiva usean turvallisuustason sisältäville sotilaallisille tietojärjestelmille. MAC:n periaatteena oli rajata pääsy kohteisiin perustuen kohteen tietosisällön arkaluonteisuuteen sekä pyynnön tekijän turvallisuustason tunnistamiseen esimerkiksi kulkuluvan perusteella. DAC soveltui taas paremmin teollisuuden ja siviilipuolen käyttöön. Sen periaatteena oli henkilökohtainen harkinta, eli järjestelmien käyttäjillä oli mahdollisuus sallia tai estää toisten käyttäjien pääsy järjestelmiin perustuen käyttäjien tietoihin tai ryhmiin, joihin kohteet kuuluvat. DAC salli käyttäjille oikeuden hyväksyä pääsy heidän hallinnassa oleviin sovelluksiin ilman järjestelmien pääkäyttäjien apua. (Ferraiolo & Kuhn 1992, 1-3.)

Ensimmäisen RBAC-mallin esittivät Ferraiolo ja Kuhn vuonna 1992. Uusi malli oli paranneltu ja yhdistetty versio aiemmista MAC- ja DAC-malleista. MAC oli liian vaikea käyttää suurten käyttäjämäärien kanssa, ja DAC oli liian jäykkä malli moderneille sovelluksille. DAC:n ongelmaksi muodostui henkilöillä annettu liiallinen harkintavalta käyttöoikeuksista, sillä käyttäjillä ei tulisi olla liikaa valtaa tietoon, minkä he omistavat. RBAC mahdollisti joustavamman ja tietoturvallisemman tavan hallita järjestelmien käyttöoikeuksia keskitetysti. RBAC-mallin avulla voitiin rajata käyttöoikeuksia myös toimintoihin eikä pelkästään suoraan tietoihin, kuten DAC- ja MAC-malleissa aiemmin. RBAC:n myötä ajattelumalli käyttöoikeuksien myöntämisestä muuttui. Tärkeää ei ollut määrittää mihin tietoihin oli oikeus vaan mitä toimintoja millekin tiedolle pystyi tekemään. Roolien ajateltiin olevan nippu tapahtumia, joita käyttäjät tai käyttäjäryhmät voivat organisaatiossaan tehdä. (Ferraiolo & Kuhn 1992, 1-4.)

RBAC-mallin ensisijaisena tarkoituksena oli helpottaa järjestelmien käyttöoikeuksien hallintaa. Käyttöoikeuksia ei jaettaisi enää yksittäisinä asetuksina vaan roolien kautta. Uuden henkilön tullessa töihin tai työtehtävien muuttuessa henkilöllä

asetettaisiin yksinkertaisesti uusi rooli, tai henkilön työsuhteen päättyessä rooli poistettaisiin. (Ferraiolo & Kuhn 1992, 7.)

Uusi RBAC-malli koostui henkilöistä, rooleista ja transaktioista eli tapahtumista. Tapahtumilla tarkoitettiin yhtä tai useampaa toimenpidettä, joita voitiin tehdä kohdejärjestelmiin. Periaatteen mukaan samalla henkilöillä voi olla yksi tai useampi rooli. Jokainen rooli voidaan valtuuttaa suorittamaan yhtä tai useampaa tapahtumaa. Periaatteiden mukaan henkilöllä on oikeus suorittaa tapahtumia vain, jos henkilölle on asetettu kyseinen rooli. Lisäksi roolit tulee vahvistaa tietylle henkilölle, ja henkilö voi suorittaa tapahtuman, vain jos tapahtuma on vahvistettu aktiiviselle roolille. Näillä säännöillä haluttiin varmistaa se, että henkilö voi saada vain niitä rooleja ja voi suorittaa vain niitä tehtäviä, mihin heillä on oikeus. Roolit voitiin myös muodostaa toisista rooleista, joten roolit oli mahdollista järjestää hierarkkisesti. Kuviossa 6 esitetään roolien periaatemalli, jossa objektit voidaan liittää tapahtumien välityksellä rooleihin ja roolit käyttäjiin. Kuviossa esitetään myös roolien hierarkkiset suhteet (Ferraiolo & Kuhn 1992, 4-8.)



KUVIO 6. Roolien hierarkkinen periaate (Ferraiolo & Kuhn 1992, 8)

RBAC –mallissa esiteltiin myös kaksi roolien hallinnan myöhemmissäkin malleissa käytettyä tärkeää periaatetta. Vähäisimpien käyttöoikeuksien periaatteen (eng. Least Priviledge) mukaisesti henkilöllä tulisi olla vain ne käyttöoikeudet, joita hän välttämättä työssään tarvitsee. Periaatteen mukaisesti henkilöiltä tuli kieltää kaikki käyttöoikeudet, mitä he eivät työtehtävien suorittamiseksi tarvitse. Toinen periaate eli vastuiden rajaaminen (eng. Separation of Duties) on säännöstö käyttöoikeuksien rajaamiseksi. Vastuiden rajaamisesta käytetään myös termiä tehtävien eriyttäminen (VM 2008, 174), tai aihepiirissä käytetään yleisesti termejä vaaralliset yhdistelmät tai kielletyt yhdistelmät. Työtehtäviin kuuluvat käyttöoikeudet antavat toisinaan mahdollisuuden tehdä petoksia tilaisuuden tullen. Kiellettyjen yhdistelmien avulla voidaan estää tiettyjen käyttöoikeuksien yhdistelmien antaminen samalle henkilölle. Esimerkiksi palkkojen asettaminen ja hyväksyminen eivät ole sallittuja toimenpiteitä samalle henkilölle. (Ferraiolo & Kuhn 1992, 9.)

Vastuiden rajaaminen voidaan jakaa kahteen pääluokkaan: staattisiin sekä dynaamisiin. Staattiset kielletyt yhdistelmät asetetaan rooleille kiinteästi, ja ne vaikuttavat vain henkilöille, joille roolit myönnetään, kun taas dynaamiset kielletyt yhdistelmät huomioidaan suoritettavan operaation aikana. Dynaaminen malli mahdollistaa staattista mallia joustavamman toiminnan. Esimerkiksi edellä mainitun staattisen säännön mukaan henkilölle ei voida myöntää palkan asettajan roolia samalla kuin hyväksyjän roolia. Dynaamisen mallin mukaan roolit voidaan myöntää samaan aikaan, mutta kyseisiä toimintoja ei voi suorittaa samaan aikaan. (Ferraiolo & Kuhn 1992, 9-10.)

5.3 Laajennettu RBAC-malli 1996

Sandhu ja kumppanit esittivät neljä paranneltua RBAC-mallia vuonna 1996. Perusmalli $RBAC_0$ kuvasi minimivaatimukset, joita järjestelmien tulisi tukea RBAC:n mukaisessa pääsynhallinnassa. $RBAC_1$-malli kuvasi roolien hierarkkisia suhteita ja käyttölupien perintää. $RBAC_2$-malli kuvasi rajoitteet, joihin kuuluvat mm. kielletyt yhdistelmät ja roolien keskinäinen poissulkeminen. $RBAC_3$-malli yhdisti kaikki kolme edellistä mallia yhdeksi. (Sandhu ym. 1996.)

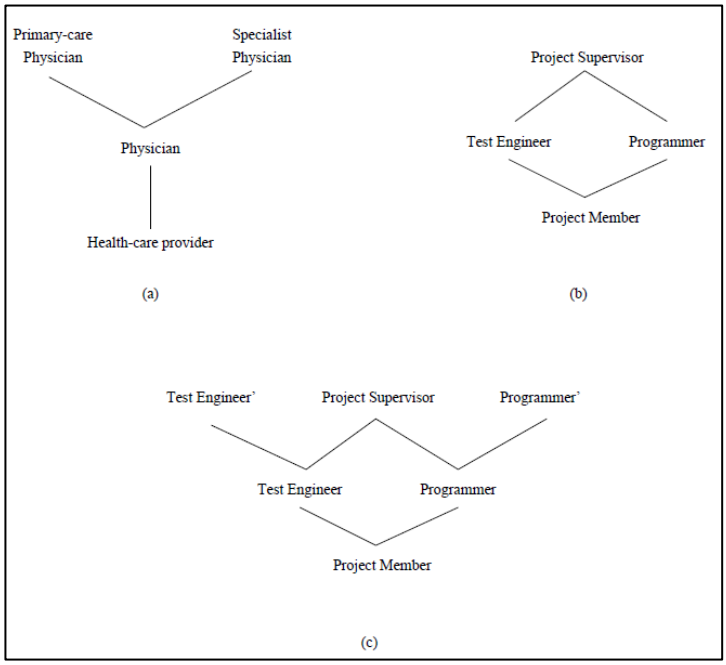
$RBAC_{0-3}$ –mallien periaatteet on koottu yhteen kuviossa 8. Seuraavissa kappaleissa kerrotaan tarkemmin jokaisen mallin periaatteet.

$RBAC_0$-malli perustui periaatteeltaan samaan kuin aiemmin Ferraiolon ja Kuhnin (1992) malli sisältäen samat perusryhmät: käyttäjät, roolit ja oikeudet. Näiden lisäksi mukaan tuotiin käsite istunto. Periaatteen mukaan käyttäjiä voitiin liittää rooleihin, ja rooleille voitiin asettaa oikeuksia. Monesta moneen suhteet säilyivät edelleen, eli rooli voitiin asettaa usealle käyttäjälle ja käyttäjällä pystyi olemaan useampi rooli ja sama koski myös roolien ja lupien välisiä suhteita. Uutena käsitteenä malliin lisättiin istunto, jonka avulla käyttäjä yhdistetään rooliin. Kun käyttäjä aktivoi roolin, käyttäjän ja roolin välille syntyy istunto. Yhtä käyttäjää kohden voi olla auki useita eri istuntoja, joista jokainen voi sisältää oman yhdistelmän aktiivisia rooleja. Tämä ominaisuus RBAC-mallissa tuki hyvin tärkeää roolien hallinnassa käytettyä vähäisimpien käyttöoikeuksien periaatetta, jonka mukaan käyttäjällä tulisi olla vain ja ainoastaan ne käyttöoikeudet, joita työtehtävien suorittamiseen tarvitsee. Istunto-käsittelyn ansiosta rooli oli mahdollista pitää aktiivisena vain tarvittaessa. (Sandhu ym. 1996, 5-6.)

$RBAC_1$-malli esitteli roolien hierarkian, joiden avulla oli mahdollista järjestellä roolit paremmin sopimaan organisaatioiden tarpeisiin. Hierarkkisen mallin perusperiaate on yksinkertainen: Rooli voi sisältää toisia rooleja, jotka perivät ylemmän tason roolien käyttöoikeudet. Mallin monimutkaisuutta voidaan selventää oheisten kuviossa 7 esitettyjen esimerkkiskenaarioiden avulla. Skenaariossa (a) rooli *physician* perii oikeudet roolilta *health-Care provider*. Roolit *primary-care physician* ja *specialist physician* kumpikin perivät roolin *Physician* sisältämät oikeudet, mutta kummallekin voidaan asettaa lisäksi suoraan omia tapauskohtaisia oikeuksia. (Sandhu ym. 1996, 8-9.)

Skenaariossa (b) on esitetty perintä, jossa on haluttu rajoittaa tiettyjen roolien perittäviä oikeuksia. Roolit *test engineer* ja *programmer* perii perusoikeudet roolilta *project member*, ja tämän lisäksi kummallakin roolilla on omat yksilölliset oikeudet. Rooli *project superior* perii kaikki roolit. Skenaariossa (c) on haluttu luoda yksilöllinen erikoisrooli *test engineer'*. Kyseinen rooli perii roolin *test engineer*

oikeudet, mutta sisältää lisäksi yksilöllisiä erikoisoikeuksia. Tapauksessa on kuitenkin haluttu erottaa se, että rooli *project superior* ei peri näitä erikoisroolien sisältämiä oikeuksia, joita *test engineer'* sisältää. (Sandhu ym. 1996, 8-9.)

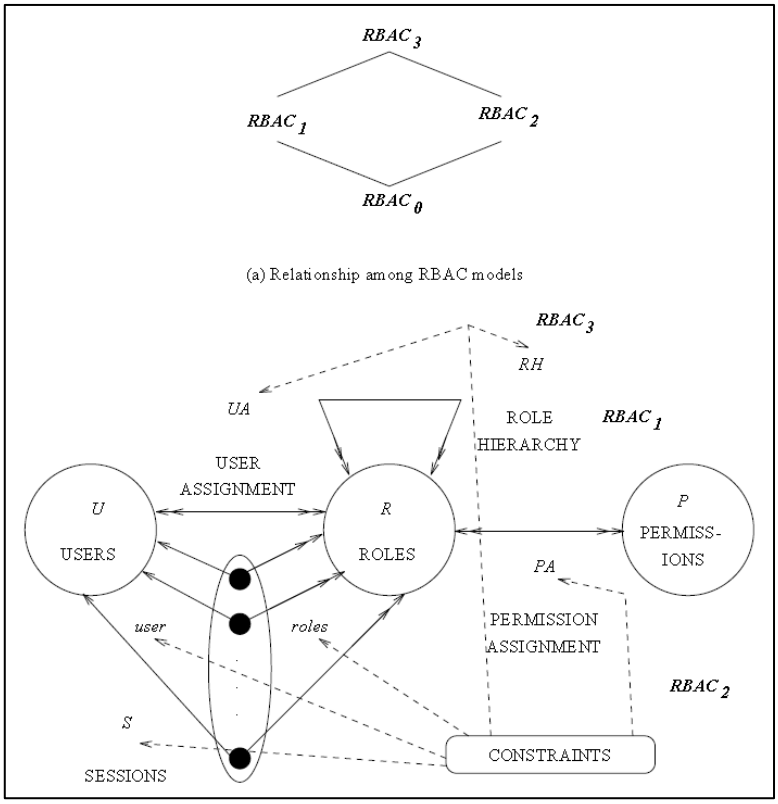


KUVIO 7. Roolihierarkioiden esimerkkiskenaariot (Sandhu ym. 1996, 19)

$RBAC_2$-malli kuvaa rooleihin liittyvät rajoitukset, joilla voidaan rajata rooleihin liittyviä käyttöoikeuksia. Useimmin käytetty rajoite RBAC-maailmassa on roolien keskinäinen poissulkeminen (eng. mutually exclusive roles), ja tähän voidaan käyttää jo aiemmin Kuhnin vuoden 1992 mallissa esitettyä *vastuiden rajaamisen* (separation of duties)-mallia. Periaatteen mukaisesti voidaan määrittää joukko rooleja, joista käyttäjä voi saada vain yhden kerrallaan. Esimerkiksi käyttäjällä voi olla roolit *ohjelmoija* tai rooli *testaaja* eri projekteissa, mutta samassa projektissa henkilöllä voi olla vain toinen rooleista. Rajoitteena voi toimia myös rooleille määrätty maksimimäärä käyttäjiä, joka kertoo, kuinka paljon käyttäjiä käyttää tiettyä roolia samaan aikaan. Rooleihin voi liittyä myös riippuvuuksia, jolloin esimerkiksi roolin A myöntäminen voidaan estää, ellei henkilöllä ole jo ennestään tiettyä roolia B. Esimerkiksi roolia *erikoislääkäri* ei voi myöntää, ellei henkilöllä ole ennestään rooli *peruslääkäri*. (Sandhu ym. 1996, 11-12.)

Samoja periaatteita voidaan myös soveltaa roolien sisältämiin käyttöoikeuksiin. Käyttöoikeus voi kuulua vain yhteen rooliin kerrallaan tai käyttöoikeutta ei voi liittää rooliin, ellei roolissa ole ennestään tiettyä käyttöoikeutta. Myös istunnoissa voidaan käyttää rajoitteita. Henkilöllä voi olla useita rooleja, mutta useammat roolit eivät voi olla aktiivisia yhtä aikaa. Samoin voidaan rajata istuntojen maksimimäärä, mitä henkilöllä voi olla samaan aikaan voimassa tai voidaan rajata istuntojen määrä, kuinka moneen istuntoon tietty käyttöoikeus voi kuulua.

$RBAC_3$-malli yhdistää edellä kuvattujen mallien $RBAC_0$ – $RBAC_2$ periaatteet yhteen. Malli sisältää aiemmin kuvatut perus-, hierarkia- ja rajoitemallin. Eri mallien yhdistäminen on tuonut mukanaan myös vaikeuksia. Jos esimerkiksi rajoituksena henkilöllä saa olla vain yksi rooli kerrallaan aktiivisena, tulee huomioida koskeeko rajoitus myös perittyjä rooleja. (Sandhu ym. 1996, 12-13.)



KUVIO 8. Roolien mallit $RBAC_{0\text{-}3}$ samassa kuvassa (Sandhu ym. 1996, 18)

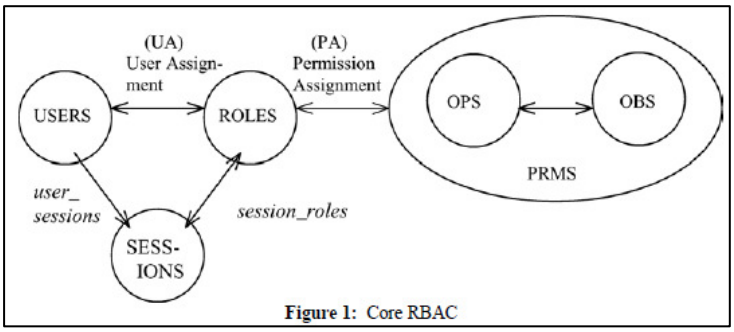
5.4 RBAC-standardi ANSI INCITS 359-2004

Ensimmäisen yhdenmukaisen standardin kehittämisen informaatioteknologian ja Yhdysvaltojen hallituksen tarpeisiin aloitti National Institute of Standards and Technology (NIST). Eri sovellustoimittajat olivat alkaneet toteuttaa aiemmin esitettyihin RBAC-malleihin liittyviä toimintoja tietokanta-, tietoturva- ja käyttöjärjestelmäsovelluksiinsa ilman yhteistä sopimusta, kuinka RBAC tulisi määritellä, ja tähän NIST haki yhteistä ratkaisua. Ensimmäinen luonnos standardista esiteltiin vuonna 2000 ACM:n (Association for Computing Machinery) tapahtumassa ja toiminnallinen määrittely julkaistiin vuonna 2001. Uusi standardi *Role Based Access Control* (ANSI INCITS 359-2004) hyväksyttiin IT- alan standardeja kehittävän yhteisön INCITS:in (The InterNational Committee for Information Technology Standards) toimesta vuonna 2004. Standardi sisälsi RBAC-mallin, jolla oli kaupallisten markkinoiden hyväksyntä. Se sisälsi yleisen mallin ja toiminnalliset kuvaukset, joita voitiin käyttää sovellusten pääsynhallinnan suunnittelussa. (ANSI INCITS 359-2004.)

Standardi perustuu neljään pääkomponenttiin, jotka olivat jo kuvattu aiemmissakin malleissa: Perus-RBAC (eng. Core RBAC), hierarkkinen RBAC (eng. Hierarchical RBAC), Staattinen vastuiden rajaaminen (eng. Static Separation of Duties) ja dynaaminen vastuiden rajaaminen (eng. Dynamic Separation of Duties). Perus-RBAC-malli kuvaa minimivaatimukset, joita järjestelmien tulee tukea. Nämä vaatimukset sisältävät roolien asettamisen käyttäjille, käyttöoikeuksien asettamisen rooleille sekä istuntojen käyttämisen roolien aktivoimiseksi. (ANSI INCITS 359-2004, 2.)

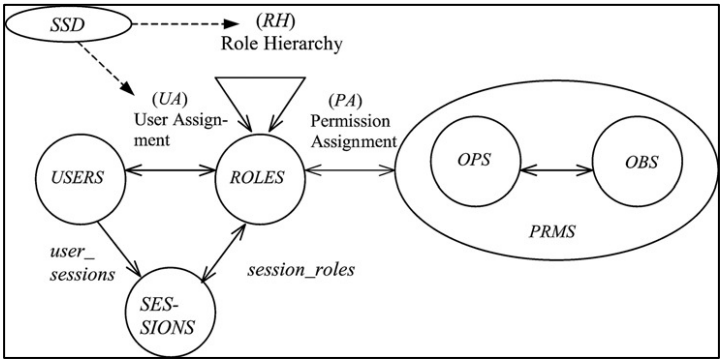
Hierarkkinen RBAC-komponentti määrittelee roolien väliset suhteet ja periytyvyyden. Staattinen vastuiden rajaaminen rajoittaa roolien asettamista käyttäjille, ja dynaaminen vastuiden rajaamisen malli rajoittaa roolien aktivointia. Nämä komponentit eivät ole pakollisia ja ne voidaan toteuttaa itsenäisinä osioina riippuen eri sovellusten tarpeista. Kuviossa 9 kerrotaan perusmallin sisältävät elementit. Käyttäjät (USERS) määritellään ihmisiksi, vaikkakin mallia voidaan soveltaa myös koneille, verkostoille tai itsenäisille järjestelmäagenteille. Roolit (ROLES) ovat

käyttäjän työtehtäviä organisaatiossa, joihin liittyy valtaa ja velvollisuuksia. Lupa (PRMS) sisältää oikeuden suorittaa hyväksyttyjä operaatioita (OPS) kohdistuen yhteen tai useampaan objektiin (OBS). Reaalimaailmassa lupa tarkoittaa esimerkiksi kohdejärjestelmän käyttöoikeusryhmää. Operaatiot kuvastavat sovelluksen toimintoja, jotka suorittavat käyttäjän valitsemia tehtäviä. Jokainen istunto (SESSIONS) linkittää henkilön yhteen tai useampaan rooliin. Toisaalta yhteen henkilöön voi liittyä yksi tai useampi istunto. (ANSI INCITS 359-2004, 2-4.)

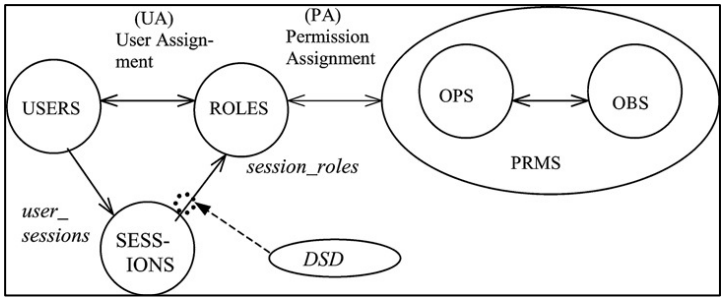


KUVIO 9. RBAC perusmalli (ANSI INCITS 359-2004, 4)

Kuvissa 10 ja 11 havainnollistetaan staattisen ja dynaamisen vastuiden rajaamisen eroja RBAC-mallissa. Staattinen vastuiden rajaaminen (Kuvio 10, SSD) kohdistuu käyttäjien ja roolien väliseen suhteeseen tai roolihierarkiaan. Kyse on kiinteistä rajoitteista, joilla estetään roolien myöntäminen käyttäjille. Dynaaminen malli (Kuvio 11, DSD) kuvastaa vastuiden rajaamista istunnon ja roolin välillä. Tämä ei estä roolien asettamista käyttäjiin, vaan se estää roolien aktivoimisen istunnon aikana. (ANSI INCITS 359-2004, 8-10.)



KUVIO 10. Staattinen vastuiden jakaminen SSD (ANSI INCITS 359-2004)



KUVIO 11. Dynaaminen vastuiden jakaminen DSD (ANSI INCITS 359-2004)

RBAC-standardi ottaa kantaa staattisiin rajoituksiin kahdella tavalla. Ensimmäisessä huomioidaan roolijoukon koko, jolla voidaan rajata sallittujen roolien määrä. Esimerkiksi käyttäjälle voi olla sallittua myöntää vain kolme roolia kerrallaan tietystä roolijoukosta. Lisäksi otetaan kantaa tiettyihin rooliyhdistelmiin, jotka estävät roolien myöntämisen samalle henkilölle. Samat rajoitukset tulee huomioida myös hierarkkisissa rooleissa. Roolien perintä ei myöskään voi rikkoa rajoituksia. (ANSI INCITS 359-2004, 8-9.)

Dynaamisten rajoitusten osalta standardissa määritellään ominaisuuksia, joiden avulla voidaan asettaa rajoituksia aktiivisille rooleille yhden istunnon sisällä tai istuntojen välillä. Henkilölle voidaan liittää kaksi tai useampi rooli, jotka eivät aiheuta kiellettyjä yhdistelmiä, jos roolit ovat aktiivisia yksittäisinä, mutta samaan aikaan aktiivisina aiheuttavat rajoitteen. Esimerkiksi jos henkilö toimessaan kassanhoitajana yrittää vaihtaa roolinsa kassanhoitajien esimieheksi, dynaamiset rajoitteet estävät vaihdon tai vaativat sulkemaan ensiksi aiemman kassanhoitajan roolin. (ANSI INCITS 359-2004, 9-10.)

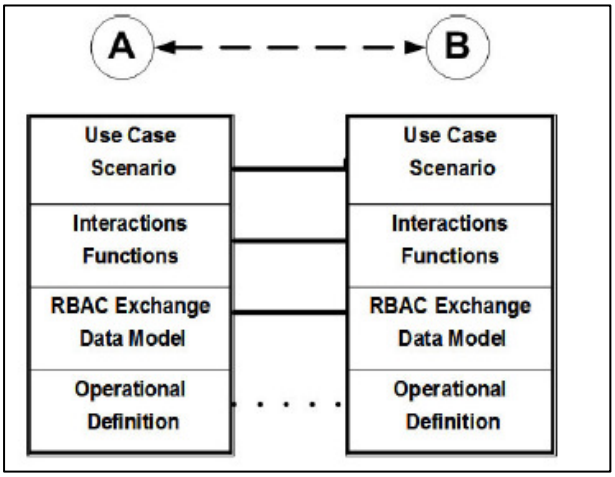
RBAC-standardissa kuvataan myös käsitemallin lisäksi yleiset funktiot joiden avulla sovellusten RBAC:n mukaista pääsynhallintaa tulisi käsitellä. Funktiot voidaan jakaa kolmeen luokkaan: Hallinnolliset funktiot kuvaavat, kuinka RBAC-elementit luodaan. Järjestelmäfunktioilla kuvataan, kuinka tietojärjestelmien ja RBAC:n välinen vuorovaikutus toimii esimerkiksi istuntojen osalta. Katselmointifunktioilla käsitellään elementtejä, joita hallinnollisilla funktioilla on luotu. Esimerkiksi RBAC-standardissa määritetty funktio *session_role* palauttaa roolit, joita

aktiivinen istunto sisältää ja funktio *session_users* palauttaa käyttäjän, joka liittyy kyseiseen istuntoon. Listaus standardissa määritellyistä funktioista löytyy liitteestä 1. (ANSI INCITS 359-2004, 6.)

5.5 Uusi paranneltu Standardi RIIS 2009

The InterNational Committee for Information Technology Standards (INCITS) on kehittänyt RBAC-standardista uutta paranneltua versiota. Aiempi RBAC-standardi oli käyttökelpoinen määrittelyyn, mutta se ei soveltunut hyvin ohjeeksi toteuttajille tai standardia arvioiville tahoille, eikä siinä myöskään huomioitu järjestelmien välistä yhteistoimintaa. RBAC-standardin akateeminen luonne karkotti käyttäjiä ja RIIS-standardin olikin tarkoitus tehdä mallista helpommin lähestyttävä ja sitä myöten auttaa käytännön toteutuksissa. Uudesta standardista *RBAC Implementation and Interoperability Standard* (RIIS) on esitelty luonnos vuoden 2009 elokuussa, ja tutkimuksen tekemisen aikana ei tätä standardia ollut vielä hyväksytty virallisesti. (Coyne 2008, 17.)

RIIS-standardi antaa opastusta RBAC:n elementtien roolinimien, lupien, hierarkioiden ja rajoitusten paketoimiseen. RIIS määrittelee mekanismit ja rajapinnat, joiden avulla voidaan siirtää RBAC-määrittelyt järjestelmästä toiseen. Tämä antaa mahdollisuuden vertailla eri RBAC-toteutuksia, jotka pohjautuvat RIIS-malliin. RIIS sisältää standardin terminologian RBAC-järjestelmien komponenteille. Jokaisessa sovelluksessa RIIS-malli jakautuu neljään osa-alueeseen: käyttötapausskenaarioihin, vuorovaikutusfunktioihin, RBAC-tiedonsiirtomalliin ja toiminnalliseen määritelmään. Rajoituksena huomioidaan se, että RIIS-mallissa ei kuvata käynnissä olevien järjestelmien välistä ajonaikaista yhteistoiminnallisuutta. Järjestelmien välisen yhteistoiminnan osa-alueet on havainnollistettu kuviossa 12. (Coyne 2008, 18-21.)



KUVIO 12. RIIS yhteistoimintamallin osa-alueet (Coyne 2008)

Järjestelmien toteuttajille RIIS-standardi ei anna yksityiskohtaisia ohjeita toteutukseen muutamaa esimerkkiä lukuun ottamatta. Toteuttajille tarkoitetut vuorovaikutusfunktiot on listattuna tarkemmin liitteessä 2. (Coyne 2008, 22-23.)

5.6 RBAC-Laajennukset

5.6.1 Yleistä

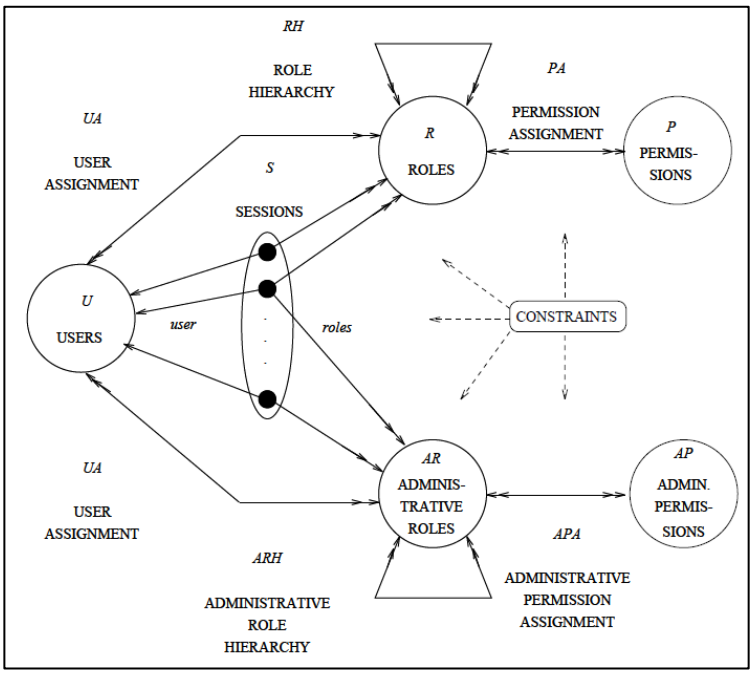
RBAC-standardista on tehty useita erilaisia muunnoksia joko laajentaen tai supistaen standardissa kuvattuja ominaisuuksia. RBAC-mallien tarkoitus on helpottaa käyttöoikeuksien hallintaa, mutta suurissa organisaatioissa ja laajoissa järjestelmäkokonaisuuksissa satojen tai tuhansien roolien ylläpito ja hallinta on vaikeaa. RBAC-mallit tukevat yhden järjestelmän pääsynhallintaa. Tästä syystä on tullut tarve kehittää muunnoksia tai laajennuksia RBAC-malliin. Seuraavissa kappaleissa esitellään muutamia tärkeimpiä muunnoksista.

5.6.2 ARBAC

Sandhu esitti vuonna 1996 uuden ARBAC-mallin (Administrative RBAC), jossa RBAC-mallia voitiin käyttää itsessään RBAC-roolien ylläpitoon. Järjestelmien

suunnittelussa voitiin huomioida se, että roolien lisääminen, poistaminen ja roolien sisältämien käyttöoikeuksien muutokset voidaan hallita samojen RBAC-periaatteiden mukaisesti.

ARBAC-mallissa aiemmin esitettyjen RBAC-mallien periaate säilyi normaalien käyttäjien näkökulmasta. Malliin on lisätty peilikuvana järjestelmien ylläpitoon liittyvät roolit ja niihin liittyvät käyttöoikeudet. Lisäksi voitiin määrittää hierarkia ja rajoitukset samaan tapaan, kuten RBAC-standardissa. (Sandhu ym. 1996, 13-14.)



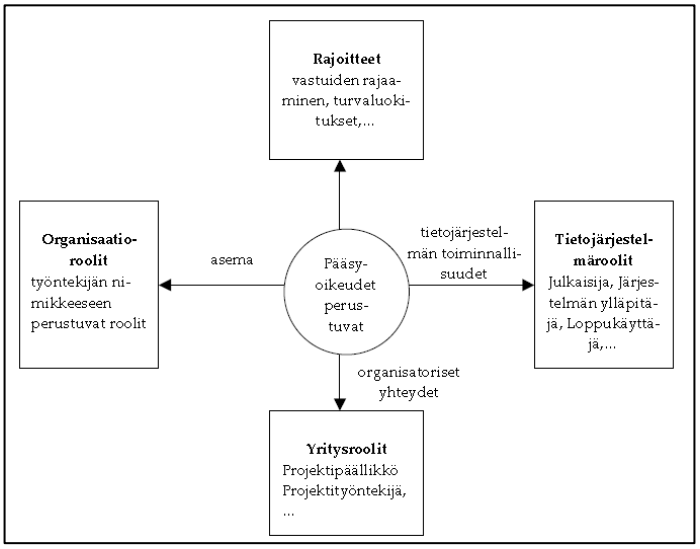
KUVIO 13. ARBAC-mallin periaate (Sandhu ym. 1996)

5.6.3 Komposiittimalli

Vuonna 2004 esitetty laajoja organisaatio- ja tietojärjestelmäkokonaisuuksia varten suunniteltu komposiittimalli oli RBAC-mallin laajennus, jossa malliin lisättiin roolien ryhmittelyyn liittyviä näkökulmia. Malli perustui organisaatioihin ja tietojärjestelmiin liittyvien roolien eriyttämiseen. Roolit jaoteltiin kolmeen pääluokkaan: organisaatioroolit, yritysroolit ja tietojärjestelmäroolit. Lisäksi mallissa

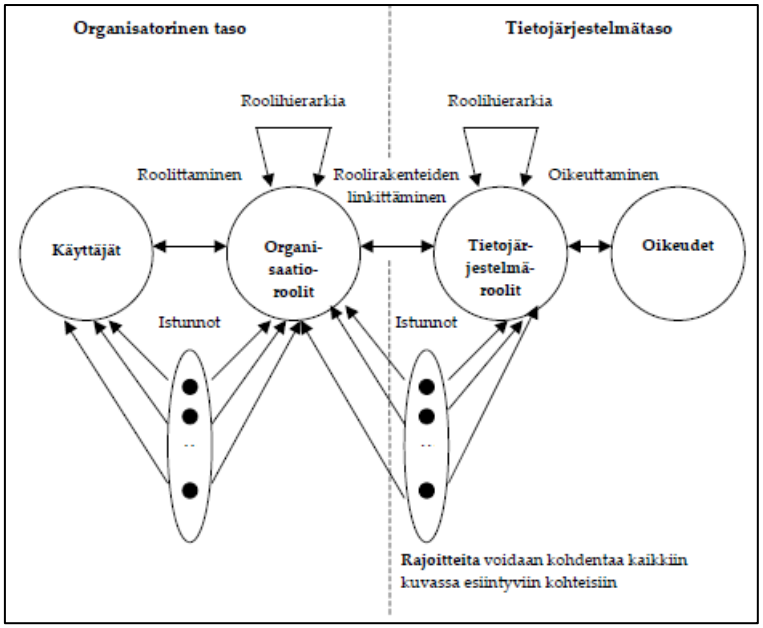
huomioitiin jo aiemmissakin RBAC-malleissa määritellyt roolien myöntämiseen liittyvät rajoitteet, kuten esimerkiksi vastuiden rajaaminen. (Mäkelä 2008, 28-29.)

Komposiittimallissa organisaatioroolit perustuvat henkilön asemaan ja ne määritetään yritysten organisaatiokaavioiden mukaisesti. Jäykkään hierarkiaan perustuvat organisaatioroolit eivät tue organisaatioiden yli tapahtuvaa työskentelyä, mistä erilaiset projektitehtävät toimivat hyvänä esimerkkinä. Tämä toi tarpeen luokitella erikseen yritysroolit. Tietojärjestelmäroolit tarkoittavat eritasoisia pääsynhallinnan tasoja, joita tietojärjestelmien käyttäminen vaati. Tietojärjestelmärooleja ovat esimerkiksi pääkäyttäjä, sisällön tuottaja ja loppukäyttäjä. (Mäkelä 2008, 29-30.)



KUVIO 14. Roolien ryhmittely komposiittimallissa (Mäkelä 2008)

Komposiittimallissa järjestelmiin liittyvät käyttöoikeudet yhdistetään tietojärjestelmärooleihin ja tietojärjestelmäroolit sidotaan käyttäjille asetettaviin yritys tai organisaatiorooleihin. Kuviossa 15 esitetään komposiittimallin periaate, joka käyttää perinteisestä RBAC-mallia sekä organisaatio että järjestelmäroolien ylläpitoon. (Mäkelä 2008, 34-35.)



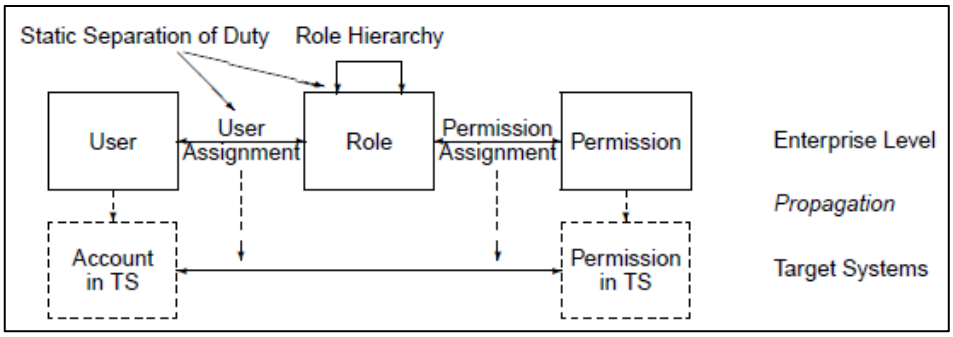
KUVIO 15. Komposiittimalli (Mäkelä 2008)

5.6.4 ERBAC

Käyttäjien ja käyttöoikeuksien hallinta isoissa yrityksissä on monimutkaista ja kallista, kun hallittavana on useita erilaisia tietojärjestelmiä. Tämä oli syy miksi kehitettiin roolien hallintaan RBAC-mallin muunnos ERBAC (Enterprise Role-Based Access Control). RBAC-malli oli rakennettu yhden järjestelmän käyttöoikeuksien hallintaa varten toisin kuin ERBAC-malli, jonka avulla oli mahdollista hallita useiden tietojärjestelmien rooleja ja käyttöoikeuksia. (Kern 2002, 1-2.)

ERBAC-mallin perusperiaate pohjautuu RBAC-malliin; käyttöoikeus liitetään rooliin ja henkilöt saavat käyttöoikeudet roolien kautta. ERBAC:ssa käyttäjille asetetaan rooleja, jotka sisältävät lupia. Lupa tarkoittaa käyttöoikeutta eri kohdejärjestelmien toimintoihin. Luvat linkitetään kohdejärjestelmissä oleviin käyttäjäryhmiin, järjestelmärooleihin tai yksittäisiin käyttöoikeuksiin. ERBAC ei käsittele istuntoja, koska eri tietojärjestelmien välillä ei yleensä ole yhteistä istuntojen hallintaa. Istuntoja hallitaan jokaisessa kohdejärjestelmässä erikseen, mikäli kohdejärjestelmä tukee istunnon käsittelyä. ERBAC-malli sisältää RBAC-mallin mukaisen roolihierarkian sekä staattiset rajoitukset. Dynaamisia rajoitteita ei mallissa

ole, koska istuntokäsitettä ei mallissa tueta. Dynaamiset rajoitteet hoidetaan tarvittaessa kohdejärjestelmissä. (Kern 2002, 5-7.)



KUVIO 16. ERBAC-malli (Kern 2002)

ERBAC-malli oli alun perin suunniteltu kaupallista sovellusta varten, ja se on tarkoitettu toimintaperiaatteeksi käyttöoikeuksien hallintajärjestelmiin, joissa työrooleihin liittyvät käyttäjä- ja roolitiedot sekä niiden muutokset välitetään kohdejärjestelmiin automaattisesti. Kyse ei ole RBAC-standardiin verrattavasta järjestelmien pääsynhallinnan mallista.

6 ROOLIEN MÄÄRITTELY

6.1 Yleistä

Roolipohjaisen käyttöoikeuksien hallinnan suurin haaste on roolien määrittelyssä. Roolien käyttö lisää hallittavuutta, mutta roolien määrittely vaatii runsaasti työtä ennen kuin rooleja voidaan käyttää. Ennen roolipohjaisen käyttöoikeuksien hallinnan aloittamista tulee määritellä optimaalinen määrä käyttötarkoitukseensa sopivia rooleja, asettaa oikeudet rooleihin ja asettaa roolit henkilöille. Hyödyt ilmenevät vasta onnistuneen määrittelyprosessin jälkeen. Epäonnistunut määrittely voi vaikeuttaa hallittavuutta entisestään. Huonossa tapauksessa määrittelyprosessin seurauksena yrityksessä on yhtä monta roolia kuin on käyttäjiäkin. Roolien määrittelystä ja oikeuksien asettamisesta käytetään kirjallisuudessa termiä *role engineering*.

6.2 Menetelmät Top-down ja Bottom-up

Roolien määrittelyä voi tehdä karkeasti kahdella eri menetelmällä: Top-down ja Bottom-up-mentelmällä. Top-down-menetelmässä käytetään yrityksessä tai organisaatiossa jo valmiiksi luotuja organisaatiorakenteita, tietoturvapolitiikkaa, toimintojen kuvausta ja prosessikuvauksia roolien määrittelyn perustana. Roolien määrittelyn jälkeen selvitetään, mitä oikeuksia rooleihin tulee sisältyä. Käytännössä tällä tarkoitetaan henkilöiden työtehtävien kuvaamista ja sopivien roolien määrittämistä työtehtävää varten. Top-down menetelmä vaatii hyvää liiketoiminnan ymmärtämistä. (Jaideep, Vijayalakshmi & Guo 2007, 1; Mienes 2003, 12.)

Bottom-up-menetelmässä käytetään olemassa olevia oikeuksia roolien tunnistamiseen. Menetelmässä kerätään tiedot tietojärjestelmien nykyisistä oikeuksista sekä tehdään henkilökohtaisia haastatteluja, joiden perusteella päätetään, mitä rooleja

on olemassa missäkin osastossa tai ryhmässä. Bottom-up-menetelmästä käytetään myös termiä *role-mining*. Menetelmiä voidaan myös käyttää yhdistelmänä. Esimerkiksi osa rooleista voidaan määrittää käyttäen Top-down- ja osa Bottom-up-menetelmää. Toisena vaihtoehtona on esimäärittää roolit Bottom-up-menetelmällä ja jalostaa rooleja Top-down menetelmän avulla. (Mienes 2003, 12; Jaideep ym. 2007, 1.)

Bottom-up-menetelmä soveltuu hyvin yrityksille, joiden olemassa olevien järjestelmien käyttöoikeudet on suunniteltu hyvin eikä niitä haluta muuttaa. Vastaavasti Top-down-menetelmä sopii tapauksiin, jossa olemassa olevia käyttöoikeuksia voidaan ja halutaan muuttaa. Roolien määrittelyn prosessi on nopeampi Bottom-up menetelmällä, ja se mahdollistaa paremmin automatisoinnin. Peruserona on se, että Top-down-menetelmässä roolit pitää määrittää, kun taas Bottom-up-menetelmässä tunnistaa. (Mienes 2003, 12; Jaideep ym. 2007, 1.)

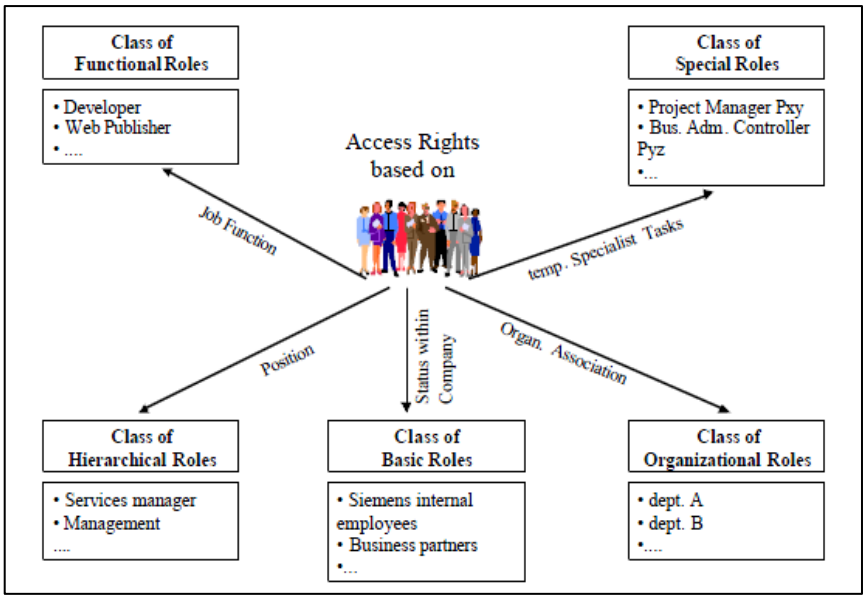
Roolien määrittämiseen Bottom-up-menetelmällä on saatavilla erilaisia kaupallisia sovelluksia. Roolien määrittely perustuu niissä erilaisiin heuristisiin menetelmiin ja algoritmeihin, joiden avulla roolit määritetään automaattisesti. Tässä tutkimuksessa ei perehdytä näihin *role-mining*-sovelluksiin tarkemmin. Tarve käyttää automatiikkaa apuna roolien määrittämisessä kasvaa sitä mukaa, mitä laajemmasta käyttöoikeuksien hallinnasta on kyse.

6.3 Prosessikeskeinen malli

Vuonna 2000 julkaistussa artikkelissa Roeckle ja kumppanit keskittyivät roolien etsimiseen liiketoiminnan prosessien näkökulmasta yritykselle, jolla oli 60 000 loppukäyttäjää. He totesivat roolien määrittelyn olevan erityisen monimutkainen asia. Liiketoimintaroolien löytäminen vaati määrittelyä tekeviltä henkilöiltä vankkaa tuntemusta määrittelyyn sisältyvien järjestelmien käyttöoikeuksien hallinnasta sekä yksityiskohtaista tietoa organisaation liiketoiminnan prosesseista. Tätä varten tuli kehittää yhtenäinen roolien löytämisen malli sisältäen roolien kuvaukset sekä

roolien etsimisen, käyttöönoton ja muutosten hallinnan periaatteet. (Roeckle ym. 2000, 1.)

Prosessikeskeinen malli perustui liiketoiminnan roolien ja varsinaisen pääsynhallinnan eriyttämiseen. Periaate tämän osalta on sama kuin Parkin ja muiden vuonna 2004 esitetyssä RBAC:n laajennuksessa komposiittimallissa (Mäkelä 2008, 28). Roolien määrittelyn selkeyttämiseksi esitettiin yleiseen käyttöön sopiva luokittelumalli, jossa roolit voitiin jakaa viiteen luokkaan: toiminnalliset roolit, erikoisroolit, organisaatioroolit, perusroolit ja hierarkkiset roolit. Luokittelun tarkoituksena oli vähentää roolien hallinnan monimutkaisuutta. (Roeckle ym. 2000, 4.)

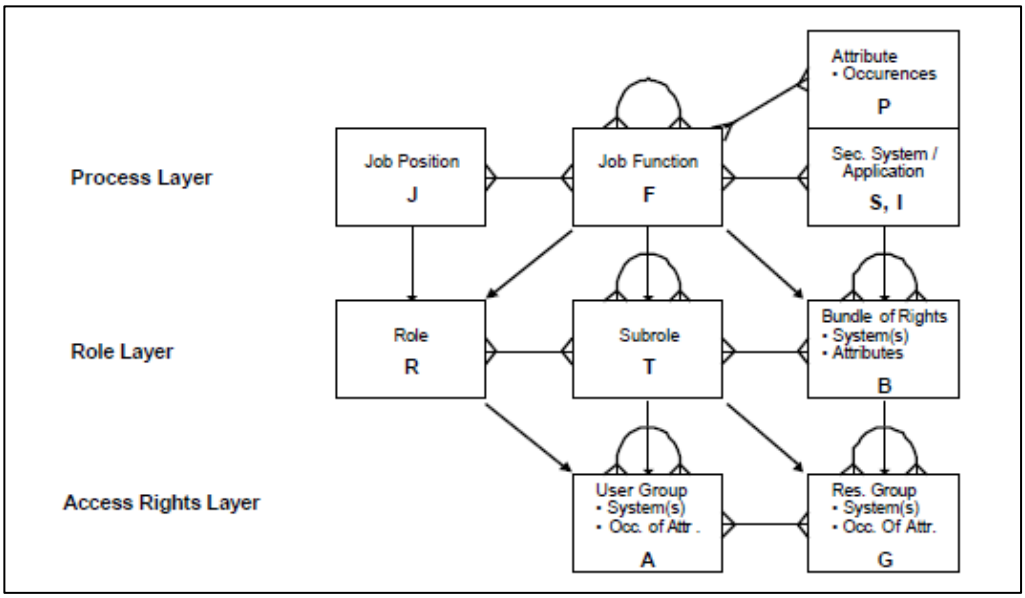


KUVIO 17. Roolien luokittelun malli (Roeckle ym. 2000)

Toiminnalliset roolit liittyivät henkilöiden normaaliin työn kuvaan, kun taas erikoisrooleilla oli tarkoitus hallita tilapäiset roolit esimerkkinä projekteihin liittyvät roolit. Organisaatioroolit pohjautuivat organisaatiorakenteeseen. Perusrooleilla oli tarkoitus hallita yrityksen henkilöstölle kuuluvia yleisiä oikeuksia. Hierarkkiset roolit perustuivat henkilöiden asemaan yrityksessä. Luokittelun hallinta oli tarkoitus toteuttaa erillisen roolikatalogin avulla. (Roeckle ym. 2000, 4.)

Roeckle ja kumppaneiden projektin tuloksena syntyi meta-malli, jonka avulla roolien löytämistä voitiin automatisoida. Malli jaettiin kolmeen kerrokseen: prosessit,

roolit ja pääsynhallinta. Prosessikerros toimi rajapintana yrityksen liiketoiminta-prosesseihin. Kerrokseen kuului manuaalisena työnä organisaatiorakenteen, työ-tehtävien ja työn kuvien määrittäminen. Prosessikerroksen tuotoksina syntyivät henkilön oikeuksiin liittyvät parametrit, joiden avulla voitiin automaattisesti aset-taa roolit henkilöille. Rooli- ja pääsynhallintakerroksen automaattiseen hallintaan käytettiin kaupallista sovellusta, joka perustui meta-malliin. (Roeckle ym. 2000, 5.)



KUVIO 18. Roolien hallinnan meta-malli (Roeckle ym. 2000, 5)

Roecklen ja kumppaneiden projektin lopputuloksena todettiin prosessikeskeisen mallin tukevan roolipohjaista käyttöoikeuksien hallintaa hyvin. Projektissa havait-tiin se, että roolien tulee perustua pääosin toiminnallisiin rooleihin. Organisaa-tiorakenteet muuttuvat, ja stabiilien roolien luominen perustuen organisaatioihin on vaikeaa. Roolien luominen perustuen olemassa oleviin työnkuvauksiin ei on-nistunut, koska työnkuvauksista puuttui näkökulma käyttöoikeuksiin. Projektissa todettiin se, että intuitiivinen roolien mallintaminen ei mahdollista perustavanlaa-tuisen roolikatalogin luomista. Projektissa luotu yhtenäinen koko yrityksen kattava roolien määrittämisen malli antoi tähän mahdollisuuden. Projektin onnistuminen perustui kahteen tekijään:

- Tietoturvapolitiikka tulee sitoa liiketoiminnan prosesseihin eikä saa polveutua pelkästään teknisestä näkökulmasta. Liiketoimintaprosessien käyttäminen auttaa ymmärtämään ja korjaamaan käyttöoikeuksiin liittyvää päätöksen tekemistä.
- Isoissa yrityksissä on välttämätöntä käyttää työkaluja roolien löytämiseen ja ylläpitoon, jotta voidaan vähentää työmäärää, parantaa hallintaa ja dokumentaatiota ja tukea liiketoiminnan tarpeita.

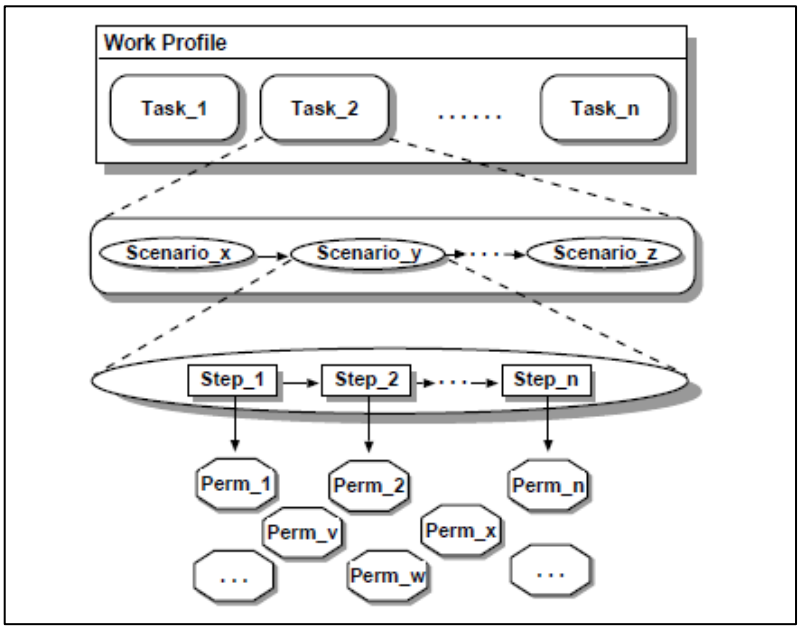
(Roeckle ym. 2000, 7)

6.4 Skenaarioihin pohjautuva malli

6.4.1 Perusmalli

Neumann ja Srembeck esittivät artikkelissaan vuonna 2002 roolien määrittelyyn mallin, joka pohjautuu työtehtävissä tapahtuviin skenaarioihin. Useat roolien määrittämisen mallit perustuivat ad hoc -tyyppiseen toimintaan, tai ne käsittivät vain osan koko prosessista. Tämän takia haluttiin luoda joustava malli, joka soveltuisi erilaisille organisaatioille ja tukisi paremmin muutosten hallintaa.

Artikkelin mukaan roolit voitiin jakaa karkeasti kahteen luokkaan: toiminnallisiin ja organisaatiollisiin rooleihin. Toiminnalliset roolit keskittyvät liiketoiminnassa tarvittaviin työtehtäviin, kun taas organisaatioroolit ovat sidoksissa yritysten organisaatiorakenteisiin. Tämä skenaarioihin pohjautuva roolien määrittämisen malli perustui toiminnallisiin rooleihin. Neumannin ja Strembeckin mallissa kuvion 19 mukaisesti henkilöiden työprofiili jaotellaan yhteen tai useampaan työtehtävään. Jokainen työtehtävistä voi sisältää yhden tai useamman skenaarion. Skenaariot voivat sisältää yhden tai useamman toimenpiteen. Näitä toimenpiteitä vastaavat järjestelmien toiminnot, mitä kohdejärjestelmässä on lupa suorittaa. (Neumann & Strembeck 2002, 1-2.)



KUVIO 19. Skenaariomallin perusperiaate (Neumann & Strembeck 2002)

Esimerkiksi työprofiililtaan lääkärin työtehtäviin kuuluu potilaiden vastaanotto. Lääkäri ottaa vastaan potilaan, jolla todetaan rintakipujen aiheuttajaksi infarkti. Skenaario voisi olla siis infarkti-potilaan tutkiminen. Toimenpiteinä potilaan tiedot tarkastetaan, tilataan näytteenottoja ja varataan aika röntgeniin. Toimenpiteisiin sisältyvät näin lupa lukea tietoja potilastietojärjestelmästä ja lupa käyttää tilaus- sekä varausjärjestelmää. (User Authorization with Role- Based Access Control 2004, 37 – 43.)

Neumann ja Strembeck laativat seitsemän portaisen mallin roolien määrittämiseksi:

- Tunnista ja mallinna skenaariot; määritetään työtehtäviin liittyvät skenaariot ja niihin liittyvät yksittäiset toimenpiteet.
- Määritä, mitä lupia skenaariot sisältää; etsitään tehtäviä vastaavat luvat eli mitä järjestelmän toimenpidettä tehtävä vastaa.
- Tunnista rajoitukset; etsitään työtehtäviin liittyvät rajoitukset. Mahdollisia rajoituksia voivat olla kielletyt yhdistelmät, aikariippuvuus, roolien keski-

näinen poissulkeminen tai terveydenhuollon rajoitukset esim. henkilökohtaisten tietojen esittäminen.

- Jalosta skenaariomalli; etsitään, ryhmitellään ja yhdistetään samankaltaiset skenaariot.
- Määrittele tehtävät ja työprofiilit; muodostetaan skenaarioista tehtäviä ja tehtävistä työprofiileja. Huomioidaan se, että sama skenaario voi kuulua useaan eri työtehtävään ja sama työtehtävä voi kuulu useaan eri työprofiiliin.
- Johda alustava roolihierarkia; järjestetään roolit hierarkkiseen järjestykseen. Mahdolliset tarpeettomat roolit kirjataan ylös myöhempää tutkimusta varten.
- Määrittele RBAC-malli; muodostetaan lopullinen kaikki roolit käsittävä RBAC-malli yhdistämällä uusi rooli aiemmin luotuihin. Yhdistetään tai erotetaan roolien hierarkiat, huomioidaan roolien sisältämät rajoitukset ja luvat ja vähennetään tarpeettomat roolit. Tämän jälkeen luodaan taas uusia rooleja ja toistetaan kunnes lopullinen RBAC-malli on valmis.

(Neumann & Strembeck 2002, 3.)

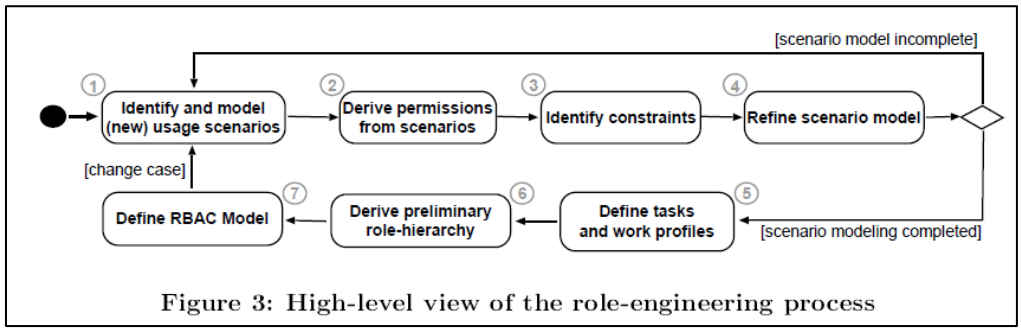


Figure 3: High-level view of the role-engineering process

KUVIO 20. Skenaariomallin vaiheet (Neumann & Strembeck 2002)

Skenaariomallin avulla tapahtuva roolipohjainen käyttöoikeuksien hallinta voidaan tiivistää myös seuraavasti:

- Mallinnetaan skenaariot, tehtävät ja luvat, jotta ymmärretään ihmisten työtehtävien sisältöä.

- Luodaan käyttöoikeussäännöt kaikille työprofiilin osille ja kutsutaan niitä termillä lupa.
- Järjestetään ja ryhmitellään lupia ja kutsutaan niitä termillä Rooli.
- Hallitaan resurssien tietoturvaa ja yksityisyyttä roolien avulla.
- Annetaan ihmisille roolit joita he tarvitsevat työtehtävissään ja päästäkseen valvottuihin resursseihin.
- Standardoidaan luvat, jotta tietoa voidaan jakaa järjestelmien ja kumppaneiden välillä.

(User Authorization with Role- Based Access Control 2004, 23.)

6.4.2 HL7

Neumannin ja Strembeckin (2002) skenaarioihin perustuvan roolien määrittämisen mallin jalostamista jatkoi Yhdysvaltojen terveydenhuollon standardeja määrittävä organisaatio HL7 (Health Level Seven). HL7 pyrki luomaan terveydenhuollon organisaatioille suunnattuja standardeja roolien määrittämisen prosessiin ja roolien määritelmiin. Pelkästä perus RBAC-mallin standardoimisesta ei ollut terveydenhuollossa tarpeeksi hyötyä, vaan tavoitteena oli vakioida RBAC-mallin elementtien nimityksiä. (HL7 2005.)

Neumannin ja Strembeckin mallin lopputuloksena syntyi roolien määritelmät. Merkittävimpänä erona tähän HL7 :n prosessimallissa standardoitujen roolien sijasta syntyy standardoituja lupia, joita voidaan liittää rooleihin. Pääasiallisena tarkoituksena oli luoda joukko yleiskäyttöisiä lupia, joita voitaisiin käyttää ”rakennuspalikoina” terveydenhuollon roolien määrittelyssä. (HL7 2005, 1.)

Määrittelyprosessin pääkohdat ovat seuraavat:

1. Skenaarioiden tunnistaminen ja mallintaminen: Jokainen skenaario kuvataan sanallisesti ja nimetään yksilöllisesti, esimerkiksi ”Luo uusi potilastietue”. Apuna voi käyttää valmiiksi tehtyjä yleisiä terveydenhuollon ske-

naariomalleja, joita ovat kuvanneet mm. HL7 ja Electronic Health Record (EHR).

2. Lupien johtaminen skenaarioista: Tunnistetaan skenaarioihin liittyvät tekijät ja vaiheet. Tunnistetaan järjestelmiin liittyvät luvat (toimenpiteet ja niitä vastaavat järjestelmäobjektit), joita vaaditaan tehtävien suorittamiseen. Listataan skenaariot ja luvat lupakatalogiin.

3. Rajoitusten tunnistaminen: Rajoitusten tunnistaminen ei kuulu varsinaisesti alkuperäiseen HL7:n prosessimalliin, koska rajoitusten on ajateltu liittyvän enemmän rooleihin, ei lupiin. Rajoitukset tullaan huomioimaan tulevaisuudessa, joten vaihe kuvataan osaksi prosessia.

4. Skenaariomallin jalostaminen: Skenaariot tarkastetaan ja määritetään tarvittaessa aliskenaariot monimutkaisimmille skenaarioille. Samankaltaiset skenaariot yhdistetään ja ylimääräiset poistetaan.

5. Tehtävien ja työprofiilien määrittäminen: Samankaltaiset skenaariot yhdistetään työtehtäviksi ja työtehtävät yhdistetään työprofiileiksi.

6. Alustavan roolihierarkian määrittäminen: Roolien määrittäminen ei ole osa HL7:n prosessia ja suositellaankin tehtäväksi jokaisessa organisaatiossa erikseen. Tulevaisuudessa on kuitenkin mahdollista määrittää yleisiä organisaatioiden välisiä rooleja, joten vaihe kuvataan osana prosessia.

(HL7 2005, 1-13.)

HL7 työn tuloksena on syntynyt vuonna 2008 julkaistu standardi *Role-based Access Control Healthcare Permission Catalog*, jossa määritellään rooleihin ja skenaarioihin liittyvät yleiskäyttöiset luvat. Yllä mainittu HL7 roolien määrittämisen prosessi ei ole vielä päässyt standardin tasolle ja on vuonna 2009 luonnosvaiheessa. Tuorein julkaisu *HL7_RBAC_Role_Engineering_Process_v1_1* on vuodelta 2005. Lisäksi vuonna 2008 on julkaistu Veterains Health Administration (VHA)

organisaation työstämä rajoitemalli *VHA Role Based Access Control (RBAC), Security and Privacy Constraint Catalog* , joka on edelleen vuoden 2010 alussa luonnosvaiheessa ja tätä ei ole hyväksytty standardiksi. (HL7 Standards 2010; HL7 DSTU 2010.)

6.5 20/80-sääntö

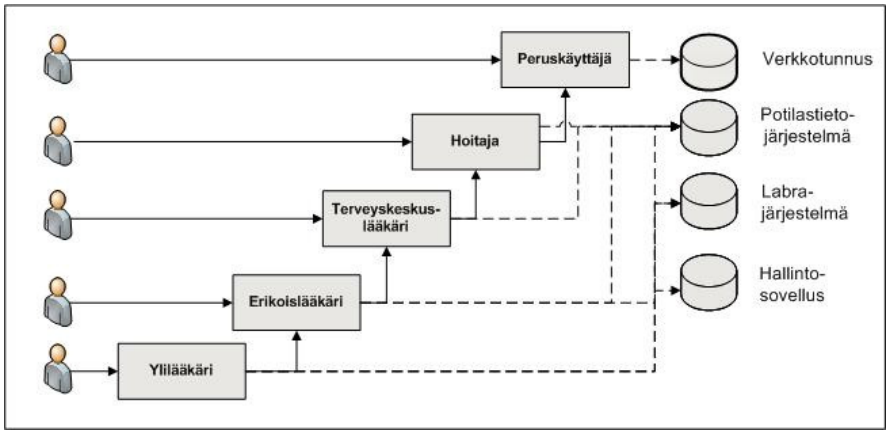
Roolien määrittely ja ylläpito on sitä vaikeampaa ja monimutkaisempaa mitä suuremmista käyttäjämääristä on kyse. Tämän takia ei ole välttämätöntä tavoitella 100-prosenttisesti kattavaa käyttöoikeuksien hallintaa roolien avulla. Mienesin artikkelin (2003) mukaan 20/80-sääntö soveltuu roolipohjaisiin toteutuksiin. Säännön perusteella 20-prosenttinen ratkaisu tuottaa 80-prosenttisen vastineen investoinnille.

Käytännön projektissa on havaittu se, että 20/80-sääntöä voidaan soveltaa myös roolien määrän optimointiin. Säännön mukaan 20 prosentilla kaikista rooleista hallitaan 80 prosenttia kaikista käyttöoikeuksista. Tämän säännön perusteella tulisi ensisijaisesti panostaa tärkeimpien roolien määrittelyyn, joiden oikeellisuus tuo parhaan hyödyn käyttöoikeuksien hallintaan. 20/80-säännön toteuttamiseen vaikuttaa myös käyttöoikeuksien hallintasovelluksen ominaisuudet. Jos sovellus perustuu täysin rooleihin eikä sillä voida hallita yksittäisiä käyttöoikeuksia, on joka tapauksessa määriteltävä huolellisesti kaikki roolit. Jos järjestelmä antaa mahdollisuuden hallita roolien lisäksi käyttöoikeuksia yksittäisinä, voidaan marginaalitapaukset (20%) hoitaa erikseen ja massa (80%) roolien avulla.

6.6 Hierarkkisuus

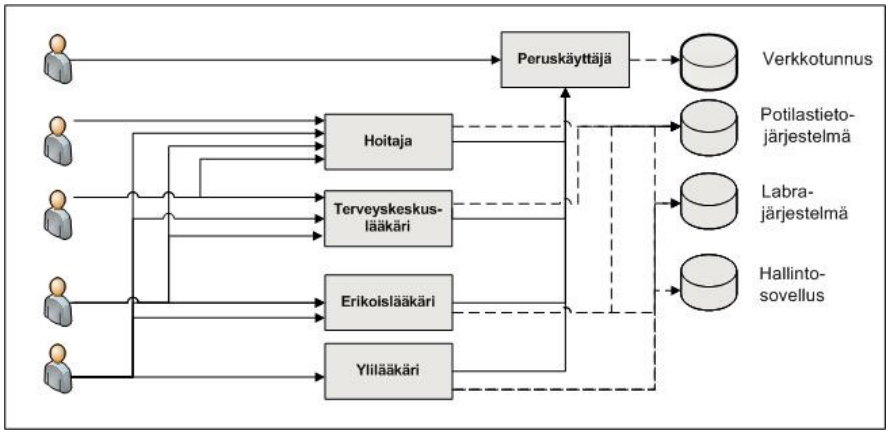
Roolien hierarkkisuus on luonnollinen tapa jäsennellä roolit tukemaan paremmin organisaation käyttöoikeuksien hallintaa. Hierarkkisuuden periaate kuvattiin tarkemmin kappaleessa 5.3. RBAC-standardien yhteydessä: rooli voi sisältää toisia rooleja, jotka perivät ylemmän tason roolien sisältämät käyttöoikeudet. (ANSI INCITS 359-2004, 5-6.)

Oheisissa kuvioissa 21 ja 22 on havainnollistettu hierarkkisuuden piirteitä yksinkertaistettujen esimerkkien avulla. Kuvion 21 rooleille on määritetty viisi eri tasoa, jossa ylemmän tason roolit perivät alemman tason roolit ja roolien oikeudet. Mallin hyvä puoli on se, että jokaiselle henkilölle asetetaan vain yksi rooli, mutta heikkoutena on joustavuus. Koska rooli pakottaa ottamaan mukaan myös aliroolit on luotava aina uusi rooli, jos haluaa erilaisen periytyvyyden.



KUVIO 21. Esimerkki roolihierarkiasta

Kuviossa 22 on esitetty sama roolijako kaksiportaisella hierarkkisuudella. Tehtävien mukaiset roolit polveutuvat kaikki roolista *Peruskäyttäjä*, mutta tehtäväroolleilla ei ole kuitenkaan riippuvuutta toisiinsa. Tällöin henkilöille on valittava useampi rooli. Malli antaa mahdollisuuden jättää pois tietty rooli, jos tarvetta roolille ei ole.

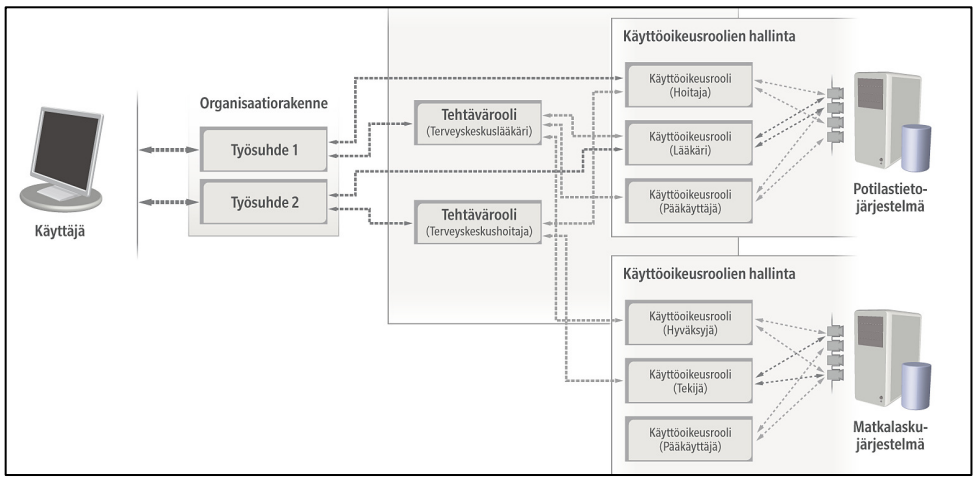


KUVIO 22. Esimerkki 2-tasoisesta roolihierarkiasta, jossa rinnakkaisia rooleja

6.7 Määrän optimointi

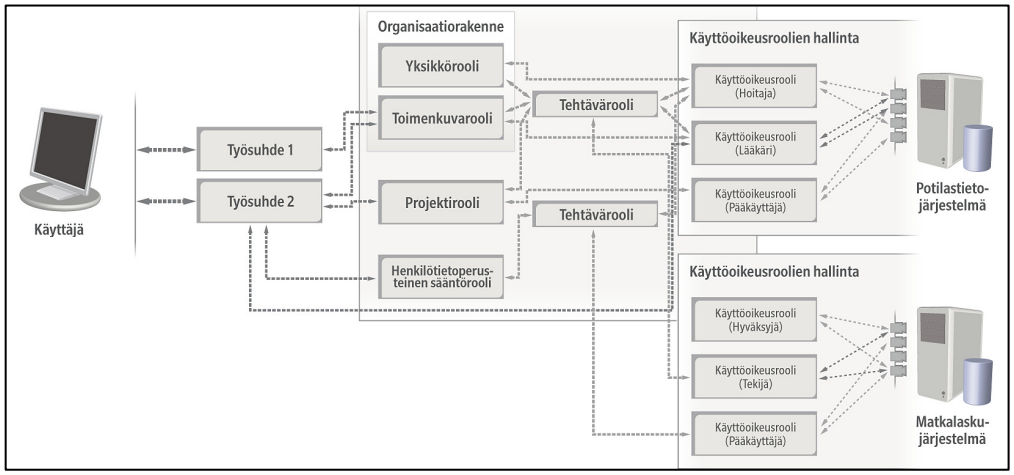
Roolien optimoinnin tavoitteena voidaan pitää määrällisesti ja laadullisesti oikean roolisetin löytämistä. Optimointia tehdään, kun organisaatiossa otetaan käyttöön roolipohjainen käyttöoikeuksien hallintajärjestelmä, mutta yhtä tärkeää on hallita roolisetin laajentaminen, kun aiemmin määritetyt roolit eivät riitä ja tulee tarve luoda uusia rooleja. Uuden roolin määrittäminen voi olla yksinkertaista, mutta se voi aiheuttaa myös ongelmia kokonaisuuden kannalta. Jos jokaista poikkeavaa tilannetta kohden luodaan uusi rooli, tulee roolien kokonaismäärä paisumaan.

Vähäisimpien käyttöoikeuksien periaatteen mukaan henkilöllä tulee olla vain hänen työtehtävissään tarvitsemat oikeudet. Tämä tarkoittaa myös sitä, että roolien sisällöksi ei voida asettaa liikaa käyttöoikeuksia. Ongelma syntyy, jos roolien määrän minimoimiseksi luodaan yleiskäyttöisiä rooleja, jotka kattavat useimpien käyttäjien tarvitsemat käyttöoikeudet ja ”varmuuden vuoksi” muutaman ylimääräisen käyttöoikeuden. Tämän ratkaisu paisuttaa käyttöoikeuksien määrää, jolloin ongelmana on turhien lisenssikustannusten syntyminen. Optimoinnin vaikeutta voidaan havainnollistaa esimerkkien avulla. Kuviossa 22 käyttöoikeuksia hallitaan tehtävä- ja järjestelmäroolien avulla. Henkilöllä on kaksi erillistä työsuhdetta, joihin kumpaankin liittyy tehtävän mukaiset roolit. Tehtäväroolit on linkitetty kahden eri järjestelmän järjestelmärooleihin.



KUVIO 22. Käyttöoikeuksien hallinta tehtävä- ja järjestelmäroolien avulla (Propentus 2009)

Roolien määrittely vaikeutuu, kun rooleja lisätään ja otetaan mukaan tehtävärooli-en lisäksi yksikkörooleja, toimenkuvarooleja, projektirooleja, henkilötietoihin perustuvia rooleja ja huomioidaan vielä roolien välinen hierarkkisuus. Roolien hallinta vaikeutuu huomattavasti. Kuviossa 23 on esitetty kyseinen tilanne. Usean erilaisen roolityypin hallitseminen vaatii enemmän osaamista roolien määrittelyyn osallistuvilta henkilöiltä, ja roolin hallinnan järjestelmien tulee tukea hyvin erilaisia roolityyppejä.



KUVIO 23. Käyttöoikeuksien hallinta useiden erityyppisten roolien avulla (Propentus 2009)

6.8 Rajoitusten huomioiminen

Roolien määrittelyn alkuvaiheessa tulee selvittää yrityksen liiketoiminnasta tai tieturvapolitiikasta johtuvat roolien rajoitteet. Rajoitteet vaikuttavat olennaisesti roolien sisältöihin. Rajoitteiden merkitys korostuu hierarkkisia rooleja käsiteltäessä. Rooli ei voi sisältää alirooleja, jotka sisältävät rajoitteita vaikuttaen isäroolin käytettävyyteen.

Kappaleessa 5.3 kuvattiin rooleihin liittyviä rajoituksia. Staattisilla rajoituksilla pyritään rajoittamaan myönnettävien roolien määrää tai estämään tiettyjen roo-liyhdistelmien myöntämisen henkilöille. Dynaamiset rajoitteet eivät estä roolien

myöntämistä, mutta ne rajaavat roolien ajonaikaista käyttöä. Esimerkiksi tietyt roolit tai tietyt roolien sisältämät oikeudet eivät saa olla yhtä aikaa aktiivisia.

6.9 Dokumentointi

Dokumentointi on tärkeä osa roolien hallinnan elinkaarta. Dokumentoinnin avulla voidaan jälkikäteen todeta perusteet, millä tavalla rooli on muodostettu. Rooleista tulisi dokumentoida yleiskuvaus, tekijät, rooleihin liittyvät rajoitteet sekä liitännän muihin rooleihin. Dokumentointi helpottaa myös uusien roolien muodostamista tai entisten muokkaamista.

Roolien dokumentointia voidaan pitää yllä erillisellä manuaalisella kirjanpidolla tai käyttöoikeuksien hallintasovelluksissa on tätä varten valmiiksi kattavat toiminnot. Ohjeen *Käyttövaltuushallinnon periaatteet ja hyvät käytännöt* (VM 2006) mukaan käyttövaltuuksien hallintajärjestelmien keskeinen vaatimus on se, että sen piirissä olevia tietoja ja tapahtumia tulee pystyä raportoimaan. Lokien perusteella tulee pystyä seuraamaan rooleihin liittyviä tapahtumia ja muutoksia. (VM 2006, 26.)

7 POHDINTAA

7.1 Roolit ja vastuu käyttöoikeuksista

Rooli on keino myöntää käyttöoikeudet käyttäjille. Rooli itsessään ei ota kantaa mihin roolia käyttävällä henkilöllä on todellisuudessa käyttöoikeus ja mistä roolin myöntäjä on ottanut vastuun. Kun henkilölle myönnetään rooli, voidaan esittää kysymys, saako henkilö käyttöoikeudeksi kyseisen roolin vai itse asiassa saako henkilö käyttöoikeuden roolin sisältöön? Vastuukysymyksissä yleinen periaate on se, että esimies vastaa alaisen käyttöoikeuksista. Mutta mistä todellisuudessa esimies ottaa vastuu myöntäessään alaiselleen roolin?

Esimerkki: Henkilölle myönnetään rooli *lääkäri* ja henkilö saa sitä kautta oikeudet *potilastietojärjestelmään*, *varausjärjestelmään* ja *verkkotunnukseen*. Mitä tapahtuu, jos roolin sisältö muuttuu ja rooliin lisätään uusi oikeus esimerkiksi *röntgensovellukseen* tai poistetaan oikeus *potilastietojärjestelmään*? Jos henkilön käyttöoikeuksia tarkastellaan vuosien päästä, tiedetäänkö varmasti, mitä yksittäisiä käyttöoikeuksia rooli *lääkäri* sisälsi kyseisellä ajan hetkellä? Myöhemmin tarkasteltaessa ei riitä tietämys siitä, mikä rooli henkilölle oli myönnetty, jos ei tiedetä, mitä roolin avulla pystyi tekemään.

Merkittävä piirre roolipohjaisessa käyttöoikeuksien hallinnassa on vastuu. Mistä roolin myöntänyt henkilö todellisuudessa vastaa? Vastaako hän pelkästään oikean roolin asettamisesta henkilölle vai vastaako hän roolin sisällöstä eli siitä, mitä henkilö voi roolin sisältämien oikeuksien avulla tehdä? Jos roolin myöntäjä vastaa pelkästä oikean roolin asettamisesta, niin vastaako roolin vastaanottanut henkilö roolin sisällöstä? Erityisen hankalaksi tulee tilanne, jossa roolia muokataan. Siirtyykö vastuu tällöin henkilölle, joka muokkasi roolia? Onhan mahdollista se, että henkilö, jolle rooli on myönnetty, ei tiedä sitä, onko roolin sisältö on muuttunut.

Tai käänteisesti voiko roolia muokannut henkilö ottaa vastuun käyttöoikeuksista, jos ei tiedä, keille henkilöille kyseinen rooli on myönnetty.

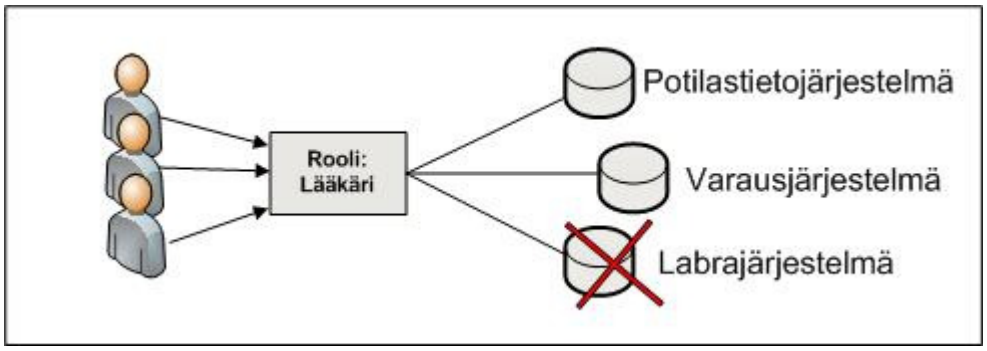
Asiaa voi tarkastella lisää edellä mainitun esimerkin kautta. Oletetaan että kyseinen lääkäri tekee rikoksen *potilastietojärjestelmän* avulla. Rikostutkimuksissa joudutaan selvittämään jälkikäteen, mitä käyttöoikeuksia henkilöllä oli ja kuka oli ne myöntänyt. Tutkimuksissa saadaan selville henkilön esimiehen hänelle myöntämä rooli *lääkäri* ja roolin sisältämät tarkemmat käyttöoikeudet mm. oikeus käyttää *potilastietojärjestelmää*. Ongelmaksi muodostuu, kuka on todella vastuussa. Onko syy esimiehessä, joka myönsi henkilölle oikeuden rooliin ja onko esimies vastuussa roolin sisällöstä, joka antoi mahdollisuuden tehdä rikoksen? Esimieshän ei välttämättä edes tiennyt, mitä käyttöoikeuksia rooli sisälsi ja voidaanko olla varmoja siitä, että rooli todella sisälsi myöntämisen hetkellä oikeuden *potilastietojärjestelmään*.

Edellä mainittu problematiikka vastuukysymyksissä tulee huomioida yrityksen tietoturvapolitiikassa. Yrityksessä tulee olla määritettynä ketkä käyttöoikeudet myöntää eri järjestelmiin ja ketkä vastaavat käyttöoikeuksien ylläpidosta (VM 2006, 11). Vastuu on myös huomioitava käyttöoikeuksien hallinnan järjestelmissä. Kaikki eri rooleihin ja roolien sisältöön liittyvien tapahtumien ja muutoksien tulee olla jäljitettävissä.

7.2 Joustavat ja kiinteät roolit

Rooleja voidaan tarkastella myös sen mukaisesti, kuinka käyttöoikeuksien hallintajärjestelmät niitä käsittelevät. Karkeasti rooleja on kahden tyyppisiä: kiinteät roolit ja joustavat roolit. Olennainen ero kiinteiden ja joustavien roolien käsittelyssä on siinä, kuinka roolit käyttäytyvät, jos niiden sisältöä muutetaan. Jos kiinteän roolin sisältöä muokataan, muuttuu kaikkien kyseiseen rooliin liitettyjen henkilöiden käyttöoikeudet. Joustava rooli antaa mahdollisuuden muokata roolia dynaamisesti käyttöoikeusprosessin aikana siten, että muutos vaikuttaa vain yhden henkilön käyttöoikeuksiin.

Kiinteitä rooleja ovat usein tietojärjestelmäroolit, joiden sisältöön ei käyttäjillä tai roolien myöntäjillä ole mahdollisuutta tai tarvetta ottaa kantaa ja joiden on tarkoitus pysyä stabiileina. Kiinteiden roolien avulla on tarkoituksen mukaista se, että sisällön muutokset astuvat voimaan nopeasti ja koskevat kaikkia henkilöitä, joille rooli on myönnetty. Kiinteät roolit sopivat pitkälle automatisoituihin käyttöoikeuksien hallinnan prosesseihin, missä roolit pysyvät stabiileina.

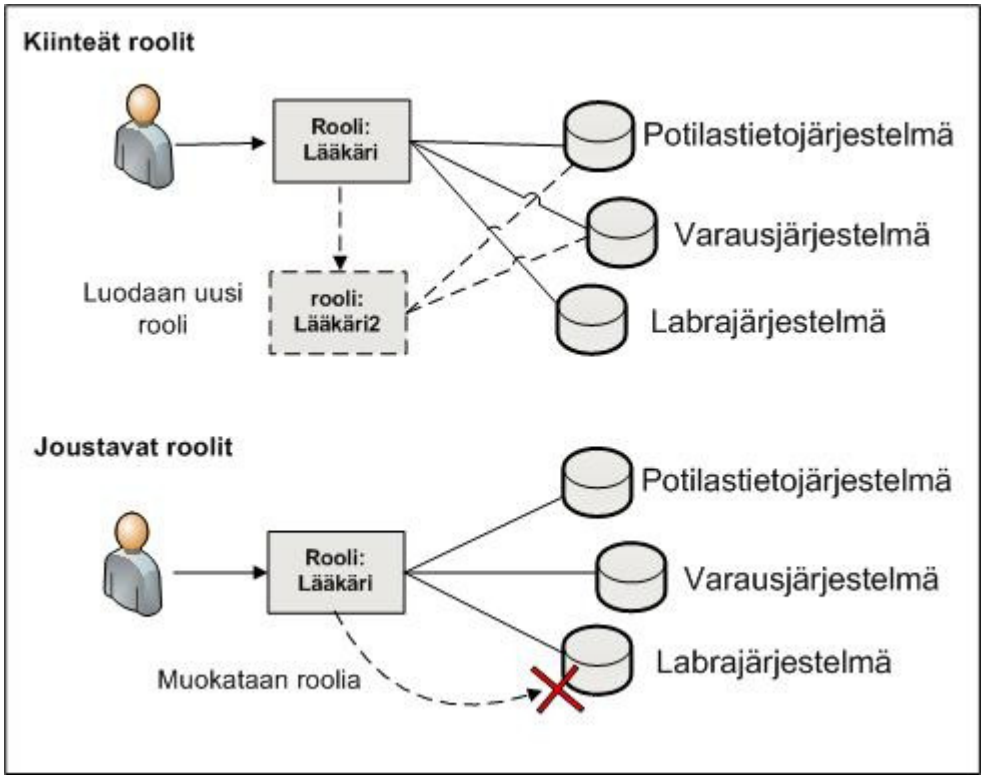


KUVIO 24. Kiinteän roolin muokkaaminen vaikuttaa kaikkiin rooliin kiinnitettyihin henkilöihin

Joustavien roolien periaate on hieman erilainen. Kun henkilölle myönnetään rooli, saa henkilö oikeastaan käyttöoikeudet roolin sisältöön eikä itse rooliin. Rooli on paketti tai nippu käyttöoikeuksia, jotka mahdollistavat usean käyttöoikeuden myöntämisen, hakemisen ja hyväksymisen yhdellä kertaa. Kun henkilölle myönnetään tietty rooli, saa henkilö joukon yksittäisiä käyttöoikeuksia, jotka on myönnetty roolin kautta. Tällöin yksittäisiä käyttöoikeuksia voidaan myös poistaa ilman sitä, että muutoksella on vaikutusta alkuperäisen roolin sisältöön. Joustavat roolit soveltuvat hyvin työrooleiksi ja erityisesti organisaatioille, joissa halutaan optimoida käyttöoikeuksien määrää käyttöoikeusprosessin aikana.

Kiinteiden ja joustavien roolin eroja voidaan tutkia esimerkin avulla. Kuviossa 25 on kuvattu sekä kiinteän roolin että joustavan roolin periaate skenaariossa, jossa henkilölle on myönnetty rooli *lääkäri*. Henkilön työnkuva muuttuu ja tarve roolin sisältämään käyttöoikeuteen *labrajärjestelmä* katoaa. Perusero kiinteän ja joustavan roolin välillä on yksinkertainen: kiinteässä mallissa poikkeavaa tilannetta var-

ten on luotava uusi rooli, kun taas joustavassa mallissa rooliin sisältöä voidaan muokata.



KUVIO 25. Kiinteät ja joustavat roolit

Kiinteässä mallissa voidaan myös muokata roolien sisältöjä, mutta tällöin muutokset vaikuttavat kaikille henkilöille, joille rooli on myönnetty. Hallinnan kannalta tämä on nopeaa ja yksinkertaista. Muutos voi olla kuitenkin vaikea, jos rooli on jo myönnetty kymmenille tai sadoille henkilöille. Roolin sisältöä muutettaessa tulee varmistaa, sopiiko muutos kaikille henkilöille, joille rooli on myönnetty. Ongelman voi kiertää helpoiten luomalla uuden roolin, kuten kuviossa 25 aiemmin esitettiin. Esimerkkitapauksessa tulisi siis luoda rooli *lääkäri2*, jolla olisi oikeudet vain *potilastieto*- ja *varausjärjestelmiin*.

Joustavat roolit antavat mahdollisuuden muokata roolin sisältöä dynaamisesti roolia myönnettäessä. Esimerkkitapauksen mukaisesti henkilöltä voidaan pudottaa pois käyttöoikeus *labrajärjestelmään,*kun tarvetta kyseiselle oikeudelle ei ole. Muutos vaikuttaa vain sille henkilölle, jonka oikeuksia muutetaan. Rooli on yhdis-

tävä tekijä, mutta ei hallitseva tekijä. Tämä tarkoittaa myös sitä että, että kaikilla *lääkärin* rooliin kytketyillä henkilöillä ei ole samoja käyttöoikeuksia. Joustavien roolien kanssa onkin olennaista käsitellä yksittäisiä käyttöoikeuksia, eikä pelkästään rooleja. Rooli kuitenkin säilyy yhdistävänä tekijänä, joka mahdollistaa muutokset myös kaikkiin rooleihin kerralla.

Joustavien roolien etuina on dynaaminen sisällön muokattavuus, jolloin turhat oikeudet voidaan jättää henkilöiltä pois, ja tämä vaikuttaa suoraan lisenssikustannuksiin. Kiinteissä rooleissa kynnys muutoksen tekemiseen on suuri, kun muutoksen vaikutus kaikkiin rooliin kytkettyihin henkilöihin tulee selvittää. Joustavat roolit eivät sovellu organisaatiolle, joissa halutaan automatisoida ja yksinkertaistaa käyttöoikeusprosessia mahdollisimman paljon. Tällöin roolien tarkoitus on usein poistaa liika joustavuus käyttöoikeusprosessista.

7.3 Yritysympäristön vaikutus käyttöoikeuksien hallintaan

Kaikille yrityksille ja organisaatioille ei sovellu samanlainen tapa hallita käyttöoikeuksia. Seuraavissa kappaleissa kuvataan, mitkä seikat yrityksissä ja organisaatioissa vaikuttavat käyttöoikeuksien hallintaan ja millä tavalla.

Yrityksen tai organisaatioiden työntekijöiden määrä vaikuttaa käyttöoikeuksien hallinnan prosessin joustavuuteen ja hallittavuuteen. Pienissä yrityksissä on mahdollista tehdä tapauskohtaisia päätöksiä käyttöoikeuksista poiketen yleisestä tavasta. Suurten käyttäjämäärien kanssa pyritään usein hyvin vakioituun ja suoraviivaiseen prosessiin, jolloin tapauskohtaiset joustot pyritään minimoimaan tai estämään niiden toteuttaminen kokonaan. Tämä periaate oli myös käytössä Roecklen ja kumppaneiden (2000) roolien määrittelyprojektissa isolle 60 000 henkilön yritykselle. Roolit tuli määrittää työtehtävien mukaan ja henkilökohtaisia neuvoteltavissa olevia käyttöoikeuksia ei sallittu.

Tietojärjestelmien ja hallittavien käyttöoikeuksien määrä asettaa vaatimuksia käyttöoikeuksien hallintajärjestelmille sekä käyttöoikeusprosessille. Pientä määrää

sovelluksia voidaan hallita yksinkertaisemmilla järjestelmillä. Suuri määrä hallittavia käyttöoikeuksia vaikeuttaa erityisesti roolien avulla tapahtuvaa käyttöoikeuksien hallintaa. Roolien määrittelyssä on tällöin kiinnitettävä huomioitava roolien hierarkkisuuteen ja määrän optimointiin. Jos hallittavana on pieni määrä käyttöoikeuksia, on roolien määrittäminen helpompaa tai käyttöoikeuksia voidaan hallita ilman rooleja. Mitä vähemmän käyttöoikeuksia on hallittavana, sitä helpompaa niiden myöntäminen, poistaminen, muokkaaminen sekä roolien määrittäminen on.

Organisaation käyttöoikeusprosessien valmiustasolla on olennainen vaikutus käyttöoikeuksien hallintaan. Tarkasti kuvattu prosessi antaa mahdollisuuden toteuttaa yhtenäistä toimintatapaa sekä seurata toimitaanko sovitun mallin mukaisesti. Jos organisaatiolla ei ole yhtenäistä prosessia, muotoutuu yrityksen sisälle erilaisia toimintatapoja ja seurattavuus heikkenee. Prosessien keskeneräisyys vaikeuttaa käyttöoikeuksien hallintajärjestelmien käyttöönottoa, koska viimeistään silloin on prosessit yhtenäistettävä.

Organisaatiorakenne määrittää, kuinka organisaatioyksikköä voidaan hyödyntää käyttöoikeuksien hallinnassa. Selkeät ja hierarkkiset organisaatiorakenteet antavat mahdollisuuden luoda sääntöjä, joiden perusteella käyttöoikeuksia tai rooleja myönnetään. Monimutkaiset ja usein muuttuvat organisaatiorakenteet vaikeuttavat automatiikan luomista.

Liiketoiminnan dynaamisuus vaikeuttaa käyttöoikeuksien hallintaa. Stabiileissa yrityksissä voidaan luoda suoraviivaisia toimintatapoja ja käyttöoikeuksia on helppo hallita, kun muutoksia ei tapahdu usein. Nykyaikaisten yritysten liiketoiminta muuttuu markkinoiden vaatimuksien mukaan. Yritykset kasvavat ja yhdistyvät, ja niiden sijainti voi muuttua. Työntekijät vaihtuvat tiuhaan ja työntekijöillä tapahtuu organisaatiomuutoksia tai työnkuvan muutoksia yritysten sisällä. Työ on usein projektiluontoista, ja projektit ovat usein yksilöllisiä. Nämä kaikki muutokset vaikuttavat suoraan henkilöiden käyttöoikeuksien hallintaan, kun eri tehtävissä ja eri organisaatioissa tulee olla erilaiset käyttöoikeudet. Tämä asettaa erityisesti

vaatimuksia käyttöoikeuksien hallinnan järjestelmille, joiden tulee tukea nopeaa muutoksen hallintaa käyttöoikeuksissa.

Yksityisten ja julkisten toimialojen käyttöoikeuksien hallinnan merkittävin ero on työsuhteiden moninaisuudessa. Yksityisellä toimialalla on henkilöllä usein yksi työsuhde. Julkisella puolella on luonteenomaista määräaikaiset työsuhteet, viransijaisuudet ja samanaikaiset virat useissa eri yksiköissä. Henkilöillä voi olla samaan aikaan useita voimassa olevia työsuhteita. Tällaisia ovat mm. opettajat ja lääkärit. Yksityiselläkin puolella käyttöoikeuksien hallintaan vaikuttaa työsuhteiden pituudet. Paljon lyhyitä määräaikaisia työsuhteita aiheuttaa paljon käyttöoikeuksien myöntämistä ja poistamista.

Julkisen sektorin tehtävistä erityisesti terveydenhuollossa on piirteitä, jotka tuovat haasteita käyttöoikeuksien ja roolien hallintaa. Sama henkilö voi toimia useissa eri työtehtävissä ja useissa eri organisaatioissa. Eri organisaatiossa voi olla samat tai eri tehtävät. Samassa organisaatiossa ja tehtävässä toimiessaan henkilöllä voi olla myös useita eri rooleja. On myös huomioitava se, että terveydenhuollon työntekijät voivat myös olla potilaina, jolloin heillä ei tulisi olla samoja käyttöoikeuksia ja pääsyjä järjestelmiin. (Ruotsalainen 2006, 64.)

Yrityksen tai organisaation hajanaisuus aiheuttaa haastetta käyttöoikeuksien hallintaa. Yhdessä sijainnissa olevalla yrityksellä voi olla yksi keskitetty käyttöoikeuksien hallinnan ratkaisu. Valvontaa ja neuvontaa voidaan tehdä tällöin paikallisesti ja tarvittaessa henkilökohtaisesti. Isoilla ja kansainvälisillä verkostoituneilla yrityksillä on usein yksiköitä eri sijainneissa kotimaassa tai ulkomailla. Käyttöoikeuksien hallinta yhtenäisellä tavalla sijainnista riippumatta vaatii tarkkaan mietityt prosessit ja prosesseja tukevan käyttöoikeuksien hallintajärjestelmän. Kansainvälisten yksiköiden osalta on myös huomioitava maakohtaiset säädökset ja lait.

Tietoturvavaatimukset ovat riippuvaisia toimialasta ja organisaatioiden tietoturvapolitiikasta. Terveydenhuolto asettaa tarkkoja määräyksiä potilastietojen käsittelyyn. Rahoitusalalla, puolustusvoimilla ja teollisuudessa on omia tarkkoja vaati-

muksia tietoturvalle. Mitä tarkemmat tietoturvavaatimukset sitä enemmän vaaditaan järjestelmiltä, joilla käyttöoikeuksia hallitaan.

Käyttöoikeustapahtumien määrä on olennainen tekijä valittaessa sopivaa käyttöoikeuksien hallintajärjestelmää. Tapahtumien määrään vaikuttavat suuresti lähes kaikki edellä kuvatut tekijät. Käyttäjien määrä, hallittavien käyttöoikeuksien määrä, tietojärjestelmien laajuus, organisaatiorakenteiden monimutkaisuus, yritysrakenteen hajanaisuus, yritysten liiketoiminnan dynaamisuus, toimialasta johtuvat työsuhteiden määrät, toimialakohtaiset säädökset ja tietoturvavaatimukset vaikuttavat kaikki käyttöoikeustapahtumien määrään. Suuri määrä tapahtumia suosii käyttöoikeuksien hallintajärjestelmiä, jotka mahdollistavat pitkälle automatisoidut prosessit. Manuaalisen työn automatisointi tuo tällöin eniten hyötyä. Stabiilit ympäristöt ja pieni määrä tapahtumia voidaan hallita hyvin suoraviivaisen ja hallitun prosessin avulla, eikä käyttöoikeustapahtumien automatisointi järjestelmien välillä ole tärkeintä. Yksinkertaistettuna voisi sanoa sen, että mitä enemmän käyttöoikeustapahtumia, sitä suurempi tarve automatisoinnille. Automatisoinnin tarpeen arvioimisessa auttaa tarkastelu, jossa verrataan käyttöoikeuksien automatisoinnin tuomaa hyötyä investoinnin kustannuksiin.

8 CASE SALO

8.1 Yleistä

Salon kaupunki tavoitteli keskitettyä ja roolipohjaista käyttöoikeuksien hallinnan prosessia ja järjestelmää, jonka avulla se pystyisi hallitsemaan n. 5000 käyttäjän käyttöoikeuksia. Käyttäjäkunta koostui kaikista kunnan ja terveydenhuollon henkilökunnasta. Projektissa otettiin huomioon vuoden 2009 alussa tapahtunut Salon seutukuntien liitos, jossa liittyi yhteen 10 kuntaa ja 4 kuntayhtymää muodostaen yhteisen Salon kaupungin. Kuntaliitoksen myötä yhdistettiin myös eri kuntien tietojärjestelmiä ja hallinnollisia tehtäviä. Myös käyttöoikeuksien hallinta siirtyi yhden yhteisen organisaation vastuulle. Tarve käyttöoikeuksien hallinnan parantamiseksi kasvoi kuntaliitoksen myötä, kun aiemmin jokaisella kunnalla oli erillisiä käyttöoikeuksien hallinprosesseja ja liitoksen myötä prosessit tuli yhtenäistää.

Salon perusvaatimuksena käyttöoikeuksien hallintajärjestelmille oli mahdollistaa henkilöstötietojärjestelmän tietoihin pohjautuva työvuo-ohjaus, jossa uusille henkilöille voitaisiin hakea ja hyväksyä rooleihin perustuvia käyttöoikeuksia sekä tarvittaessa luoda käyttöoikeuksia automaattisesti (provisioida) kohdejärjestelmiin. Järjestelmällä hallittavia käyttöoikeuksia tulisi olemaan mm. verkkotunnukset, sähköpostit, tietohallinnon järjestelmät, taloushallinnon järjestelmät, terveyden huollon järjestelmät, toimisto-ohjelmistot, kulkuluvat sekä fyysiset laitteet. Käyttöoikeuksien hallinnan piiriin haluttiin aluksi noin 15 tärkeintä järjestelmää joista ensimmäisenä otettaisiin käyttöön käyttäjätunnusten, sähköpostin, matkalaskujen, ostolaskujen ja työsopimusten hallinta.

Projektissa oli tarkoituksena luoda Salon tarpeisiin sopiva käyttöoikeuksien hallinnan malli ja kehittää samalla toimitettavan järjestelmän roolien hallinnan ominaisuuksia. Vaatimukset järjestelmälle asetettiin projektin alussa. Järjestelmässä tulisi hallita työroolien ja niiden sisältämien käyttöoikeuksien luominen, myöntä-

minen käyttäjille, poistaminen käyttäjältä, poistaminen järjestelmästä ja roolien muokkaaminen.

Projektissa käytiin läpi roolien määrittelyn perusteet järjestelmätoimittajan toimesta, mutta varsinainen määrittelytyö kuului Salon kaupungin ja kolmannen osapuolen vastuulle. Osapuolien kanssa suunniteltiin yhdessä roolien hallinnan perustoiminnallisuudet, joiden perusteella määritettiin, kuinka järjestelmän tulisi tukea roolien käsittelyä. Projektin aikana kehitettiin järjestelmän roolien hallinnan ominaisuuksia tukemaan paremmin asiakkaan tarpeita, mutta tarkoitus ei ollut tuottaa valmista roolisettiä.

8.2 Toimitettavan ratkaisun yleiskuvaus

Saloon toimitettu käyttöoikeuksien hallintajärjestelmä Propentus Permission Manager (PPM) on käyttöoikeuksien hallintaan tarkoitettu tietojärjestelmä. Sen ominaisuuksien avulla voidaan helpottaa ja tehostaa organisaation käyttöoikeuksien, kulkulupien ja laitteiden anomista, myöntämistä, poistamista sekä seurantaa. Järjestelmän tuottamat raportit auttavat seuraamaan organisaation sisäisiä prosesseja ja sääntöjä tai lakien asettamia vaatimuksia.

Järjestelmä perustuu käyttöoikeuspyyntöihin ja niiden hyväksymiseen. Käyttöoikeuden hyväksymisen vastuu on jaettu hyväksyntäketjulle, jolloin vastuu siirtyy tietohallinnolta esimiehille. Hyväksytty käyttöoikeuspyyntö siirtyy tekniselle käsittelijälle toteutusta ja kuittausta varten, tai vaihtoehtoisesti hyväksytty pyyntö provisioi automaattisesti käyttöoikeudet kohdejärjestelmiin. Tekninen käsittelijä voi olla kohdejärjestelmän pääkäyttäjä tai keskitetty helpdesk. Hyväksyjät voivat myös delegoida vastuunsa toisille hyväksyjille esimerkiksi lomien tai muiden poissaolojen ajaksi. Jokainen suoritettu toiminto jättää järjestelmään jäljen, joka voidaan myöhemmin raportoida.

Anottavissa olevia käyttöoikeuksia voidaan ylläpitää ns. tuotekatalogissa, joka perustuu yleiseen verkkokaupoissa käytettyyn ostoskorimalliin; käyttäjä valitsee

haluamansa käyttöoikeudet koriin, antaa niille tarvittavat tiedot ja lähettää ne eteenpäin hyväksyntää varten. Jokainen käyttäjä voi seurata omien käyttöoikeuksiensa tilannetta. Lisäksi esimiehet voivat seurata myös alaistensa käyttöoikeuksia ja järjestelmän omistajat heidän järjestelmiinsä kohdistuvia käyttöoikeuksia. Käyttäjien henkilötiedot voidaan tuoda suoraan HR–järjestelmistä, mutta järjestelmällä voidaan myös hallita ulkopuolisia käyttäjiä esimerkiksi konsultteja, joiden tietoja ei tallenneta HR-järjestelmiin. Järjestelmän arkkitehtuurikuva löytyy liitteestä 3.

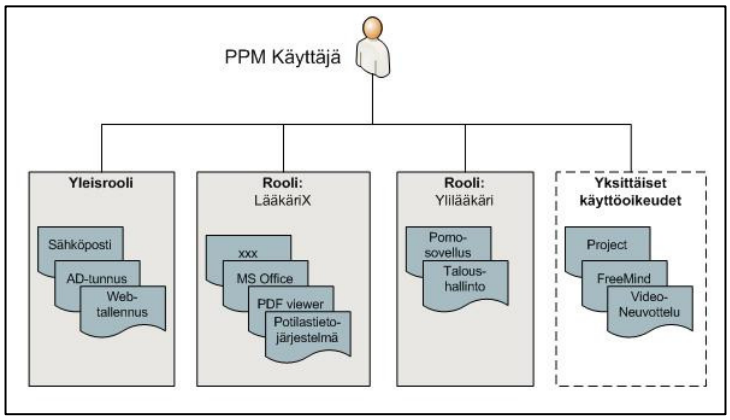
8.3 Toimintaympäristö

Salon kaupungin käyttöoikeuksien hallinnan toimintaympäristö koostuu sekä terveydenhuollon että kaupungin työyksiköistä. Kaupungin puolella käyttöoikeuksien hallinta on vakiintunutta ja pääasiassa eri järjestelmien pääkäyttäjien kautta tapahtuvaa. Terveydenhuollon puolella työ on luonteeltaan dynaamista koostuen vakinaisen henkilökunnan lisäksi suuresta määrästä määräaikaisia työntekijöitä, joiden työajat ja työsuhteiden pituudet vaihtelevat. Henkilöillä on useita lyhyitä perättäisiä työjaksoja, joista jokaiselle jaksolle tehdään erillinen työsopimus. Henkilöillä on myös useita samanaikaisia voimassaolevia työsuhteita, joista jokaisessa voi olla yksilölliset käyttöoikeudet. Käyttöoikeuksien kannalta tärkeää on nopeus ja joustavuus. Tilapäisten tuuraajien esimerkiksi lääkäreiden ja hoitajien tulee saada käyttöoikeudet lyhyellä varoitusajalla. Käyttöoikeuksien hallinta keskittyy pääosin potilastietojärjestelmän, käyttäjätunnuksen ja sähköpostin ympärille. Roolien hallinnan kannalta tärkein on potilastietojärjestelmä, jonka käyttöoikeuksia hallinnoidaan sisäisillä järjestelmärooleilla tai käyttäjäryhmillä.

8.4 Roolien määrittelyn perusteet

Salon projektissa rooleilla tarkoitettiin työrooleja. Kokonaisuudessa huomioitiin myös potilastietojärjestelmien järjestelmäroolien hallinta osana työrooleja. Projektissa ei ollut tarkoitus syventyä tarkemmin yhden järjestelmä ohjaamiseen, joten potilastietojärjestelmän järjestelmäroolien sisältöä ei lähdetty arvioimaan uudelleen. Oletuksena järjestelmäroolit olivat valmiiksi määriteltyjä.

Käyttöoikeuksien hallinnan lähtökohtana oli pyrkiä yksinkertaiseen toimintamalliin ja laajentaa mallia tarvittaessa. Käyttöoikeuksia hallinta jaettiin kolmeen ryhmään: yleisroolit, tehtäväroolit ja loppujen käyttöoikeuksien hallitsemiseen yksittäisinä ilman rooleja. Henkilölle voitiin myöntää *yleisrooli*, jonka perusteella määräytyisivät yleiset peruskäyttöoikeudet, kuten verkkotunnukset ja sähköposti. Lisäksi henkilölle valittaisiin tehtäväroolit, joiden mukaan henkilö saisi työtehtävänsä vaatimat käyttöoikeudet. Näitä rooleja olisi rajattu määrä, ja ne kattaisivat merkittävimmän osan myönnettävistä käyttöoikeuksista. Lisäksi voitaisiin hakea henkilökohtaisia oikeuksia, mikäli roolien mukaiset käyttöoikeudet eivät olisi riittävät. Roolien periaatetta on selvennetty kuviossa 26.

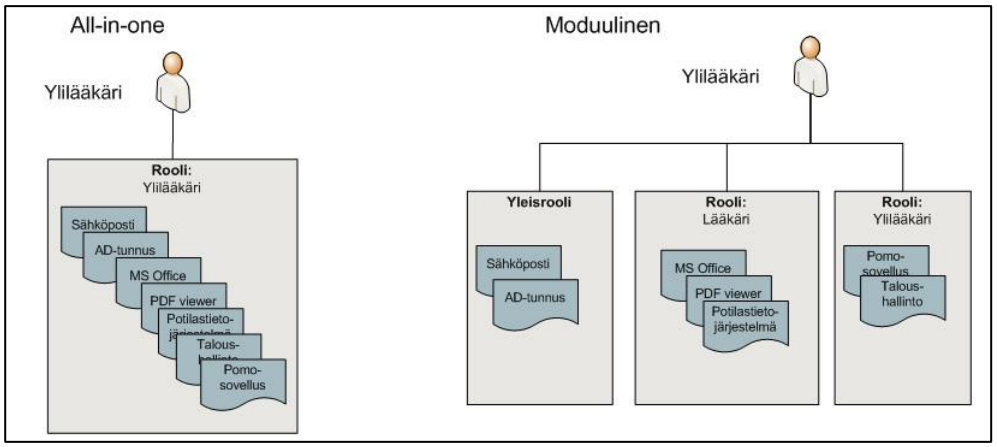


KUVIO 26. Käyttöoikeuksien hallinnan perusmalli

Roolien suunnittelussa huomioitiin seuraavat pääperiaatteet:

- Pyritään minimoimaan tarvittavien roolien määrä.
- Vältetään turhia käyttöoikeuksia; turhat oikeudet muuttuvat ongelmiksi, jos kyseessä on käyttöoikeus, joka vaatii lisenssien hankkimisen.
- Luodaan rooleja, joiden hyväksyntäketjun läpimeno on nopea; roolien sisällöksi on valittava käyttöoikeuksia, jotka voidaan hyväksyä roolin yhteydessä ja joiden oikeudet määritetään automaattisen provisioinnin avulla.
- 20/80-sääntö tulisi muistaa; kaikkia käyttöoikeuksia ei ole tarkoitus hallita roolien avulla vaan tavoitteena on löytää tärkeimmät roolit. 20% rooleista riittää kattamaan 80% käyttöoikeuksista.
- Rooleja tulee olla korkeintaan 100 kpl, jotta hallittavuus säilyy.

Roolien rakennemalleina ei koettu tarpeelliseksi lähteä toteuttamaan hierarkkista roolien hallintaa, jossa ylemmän tason roolit perisivät alemman tason roolien käyttöoikeudet. Roolien suhteen suunniteltiin lähinnä modulaarisuutta ja optimaalista roolien sisältöä. Rooleista esiteltiin kaksi rakennemallia. ”All-in-one”-mallissa jokaista haluttua roolivariaatiota kohden tehdään oma rooli, joka sisältää kaikki oikeudet, kun taas modulaarisessa mallissa roolit tulisi olla pienempiä yleiskäyttöisempiä kokonaisuuksia.

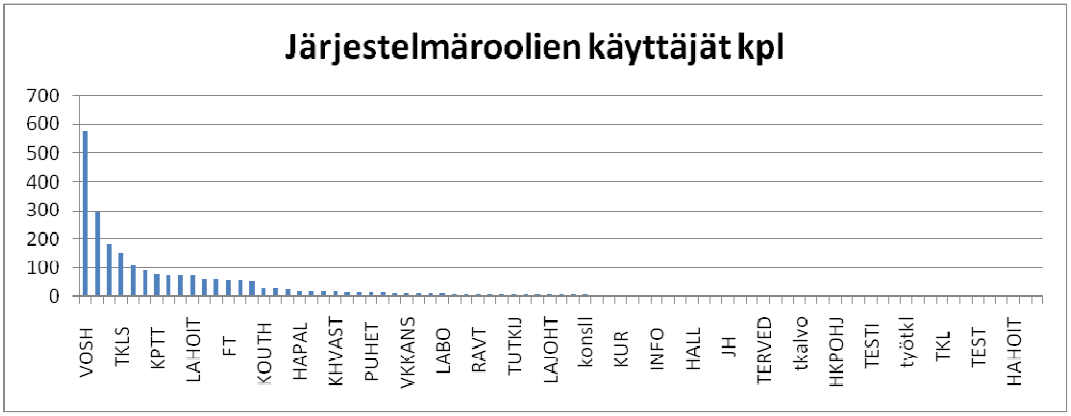


KUVIO 27. Roolien rakennemallit

8.5 Lopputulokset

Rooleista ei muodostunut projektin tärkeintä osa-aluetta, ja projektin käyttöönottovaiheessa vuoden 2010 alussa keskityttiin käyttöoikeuksien hallintajärjestelmän käyttöönottoon aluksi ilman rooleja. Järjestelmän avulla voitiin hallita roolien lisäksi myös yksittäisiä käyttöoikeuksia, ja tämä periaate antoi mahdollisuuden aloittaa järjestelmän käyttö ilman rooleja ja tuoda roolit vähitellen osaksi prosessia. Roolien mukainen käyttöoikeuksien hallinta oli projektin lähtökohta ja projektissa luotiin sille järjestelmätuki ja valmiudet.

Roolien määrittelyn yhdeksi lähtökohdaksi asetettiin keskittyminen tärkeimpien roolien määrittelemiseen 20/80-säännön periaatteen mukaisesti. Projektissa tutkittiin Effica-potilastietojärjestelmän olemassa olevia oikeuksia ja havaittiin tilastollisesti tavoiteltu periaate paikkansa pitäväksi. Kaiken kaikkiaan järjestelmässä oli 66 järjestelmäroolia, joihin oli asetettu käyttäjiä. Käyttäjiä oli järjestelmässä yhteensä 2360. Rooleista käyttäjämääriltään 13 suurinta roolia (20%) riittivät kattamaan 79,7% käyttäjistä, joten 20/80-sääntö toimi tässä tapauksessa poikkeuksellisen tarkasti. Kuviossa 28 havainnollistetaan käyttäjien jakautuminen rooleihin.



KUVIO 28. Potilastietojärjestelmän järjestelmäroolien käyttäjämäärien jakauma

Roolien määrittelyä lähdettiin alustavasti tekemään Bottom-up-tekniikalla. Tärkeimpien järjestelmien olemassa olevat käyttöoikeudet selvitettiin, ja niiden perusteella suunniteltiin alustavia rooleja. Bottom-up tekniikan hyvä puoli on se, että tekniikka paljastaa järjestelmien käyttöoikeuksien nykytilanteen. Salon tapauksessa verkkotunnuksia tutkittaessa havaittiin useita tyypillisiä ongelmia. Järjestelmässä oli voimassa tunnuksia, joita ei enää käytetä, ja henkilöillä oli käytössä useita eri tunnuksia. Yhden ongelman aiheuttavat lukuisat yhteiskäyttötunnukset, joita on luotu eri käyttötarkoituksia varten. Hyvän hallintatavan mukaan yhteiskäyttötunnuksia ei saisi olla, jotta käyttäjät voidaan aina tunnistaa.

Ongelmien löytyminen ja tunnusten siivoaminen järjestelmästä vaatii paljon työtä, ja se on tehtävä ennen roolien määrittelyä. Toisaalta tämä osoittaa sen, että käyttöoikeuksien hallinta on vaikeaa ja ongelmat kasautuvat vuosien saatossa. Roolien

mukaisen käyttöoikeuksien hallinnan pitäisi tuoda juuri tähän helpotusta. Jos tunnukset myönnetään ja poistetaan vain roolien kautta, ei ongelmia pääse syntymään. Hallintajärjestelmän avulla on myös mahdollista katselmoida tilanne ja puuttua ongelmiin ajoissa.

8.6 Projektissa kohdatut haasteet

Yksi roolien määrittelyssä kohdattu vaikeus oli yllättäen perusoikeuksien määrittäminen. Oletusarvoisesti oli tarkoitus luoda automaattinen perusrooli, joka sisältäisi kaikille käyttäjille yhteiset oikeudet. Näitä arvioitiin olevan käyttäjätunnus verkkoon, sähköposti sekä muutama tärkeä selainsovellus. Vaikka sovellukset olivat lähes jokaisen käytössä, niin tarkasteluissa selvisi myös poikkeustapauksia. Isossa organisaatiossa on aina poikkeuksia, joille ei kuulu yleisiä oikeuksia. Kuntaorganisaatiossa näitä voivat olla mm. perhe- ja omaishoitajat sekä luottamustoimia tekevät henkilöt. Poikkeustapausten hoitamista tuettiin sovelluksen joustavilla roolien hallinnan periaatteilla. Roolien myöntäminen perustuu roolin anomiseen, ja tämä antaa mahdollisuuden valita, kuuluuko henkilölle perusrooli vai ei. Järjestelmä antaa mahdollisuuden poistaa roolin sisältämiä käyttöoikeuksia dynaamisesti pyynnön aikana. Tällöin voidaan myöntää rooli ja rajata roolin sisältöä.

Projektissa kohdattiin vaikeuksia käyttöoikeuksien hallinnan käsitteiden ymmärtämisessä toimittajan ja asiakkaan välillä. Rooli on periaatteeltaan yksinkertainen, mutta eri käyttötarkoituksessa rooli tarkoittaa eri asiaa. Roolista voi olla yhtä monta käsitystä kuin on kuulijaakin. Rooleista puhuttaessa on tärkeää erottaa työroolin ja järjestelmäroolin erot. Työroolilla hallitaan kokonaisuutta ja järjestelmäroolilla yksittäisen järjestelmän oikeuksia.

Yksi projektissa kohdattu haaste oli asiakkaan tahtotilan ja toimittaja näkemyksen kohtaaminen sekä asiakkaan avainhenkilöiden näkemyserot. Projektissa tavoiteltiin lähtökohtaisesti kokonaisvaltaista ja roolipohjaista käyttöoikeuksien hallintaa. Todellinen tarve alkuvaiheessa oli tärkeimpien ydinjärjestelmien ja erityisesti potilastietojärjestelmän käyttöoikeuksien hallinta sekä peruskäyttöoikeuksien hallinta.

Projektia vaikeutti myös tavoitteiden muuttuminen projektin aikana, johon vaikutti suuresti järjestelmän suunniteltua myöhäisempi käyttöönotto. Roolipohjaisuuden tuomia hyötyjä, haittoja ja oikeaa toimintatapaa on vaikea hahmottaa ilman todellista sovellusta. Projektin pitkittyessä roolien käytännön merkitys pieneni.

Projektissa harkittiin alkuvaiheessa potilastietojärjestelmän roolien sisällön hallintaa käyttöoikeuksien hallintajärjestelmästä käsin. Suunnittelun aikana havaittiin ongelmaksi yhteisen rajapintaratkaisun saaminen potilastietojärjestelmätoimittajan kanssa projektin aikataulun rajoissa. Tämän vuoksi tehtiin päätös, ettei potilastietojärjestelmää koskevien järjestelmäroolien sisällön hallintaa oteta mukaan projektiin.

9 YHTEENVETO JA JOHTOPÄÄTÖKSET

Tutkimuksen tavoitteena oli selvittää, mitä roolipohjaisella käyttöoikeuksien hallinnalla tarkoitetaan ja mitkä tekijät vaikuttavat siihen, millainen roolipohjaisen käyttöoikeuksien hallinta sopii erilaisille organisaatioille. Työssä tutkittiin aluksi, mitä käyttöoikeuksien hallinta yleisesti on ja mitä rooleilla tarkoitetaan käyttöoikeuksien hallinnassa. Tutkimuksessa selvitettiin, mitä standardeja ja malleja rooleista on tehty ja millaisin menetelmin rooleja voidaan määritellä. Työssä pohdittiin myös rooleihin liittyviä vastuukysymyksiä sekä yritysympäristön vaikutusta roolipohjaiseen käyttöoikeuksien hallintaan.

Asiakastapauksena toimi Salon kaupunki, joka tavoitteli keskitettyä ja roolipohjaista käyttöoikeuksien hallinnan prosessia ja järjestelmää, jonka avulla se pystyisi hallitsemaan kaupungin sekä terveydenhuollon työntekijöiden käyttöoikeuksia. Projektiin kuului roolipohjaisen toimintamallin luominen ja järjestelmän kehittäminen tarpeiden mukaiseksi.

Roolien avulla pyritään tuomaan hallittavuutta, nopeutta, johdonmukaisuutta, kustannussäästöjä ja tietoturvallisuutta käyttöoikeuksien hallintaan. Kuinka roolit sitten tulisi määrittää, jotta tähän tavoitteeseen päästään? Määrittelyssä on huomioitava kaikki yritysympäristöön liittyvät tekijät ja arvioitava oikeat roolit ja oikeat menetelmät roolien määrittämiseksi. Käyttäjien määrä, hallittavien käyttöoikeuksien määrä, tietojärjestelmien laajuus, organisaatiorakenteiden monimutkaisuus, yritysrakenteen hajanaisuus, yritysten liiketoiminnan dynaamisuus, toimialasta johtuvat työsuhteiden määrät, toimialakohtaiset säädökset ja tietoturvavaatimukset vaikuttavat kaikki oikean toimintamallin valintaan. Mitä suurempi ja monimutkaisempi toimintaympäristö on, sitä vaikeampaa käyttöoikeuksien hallinta on ja sitä enemmän vaaditaan käyttöoikeuksien hallinnan järjestelmiltä.

Rooleista puhuttaessa on tärkeää erottaa työ- ja järjestelmäroolien merkitys. Työrooleilla hallitaan henkilön työtehtäviin liittyviä käyttöoikeuksia riippumatta mi-

hin järjestelmään ne kohdistuvat. Järjestelmärooleilla pyritään hallitsemaan yksittäisen järjestelmän pääsynhallintaa, ja järjestelmäroolit onkin usein rinnastettavissa järjestelmien käyttäjäryhmiin. Työrooleja voidaan luokitella tarpeen mukaan erilaisista näkökulmista. Perusrooleilla tarkoitetaan organisaation yleisiä oikeuksia, joita voi olla esimerkiksi käyttäjätunnukset ja sähköpostit, jotka kuuluvat lähes kaikille käyttäjille. Organisaatioroolit pohjautuvat organisaatiorakenteeseen, ja yksikkörooleilla voidaan hallita henkilön asemaan liittyviä oikeuksia. Tarvittaessa voidaan luoda erikoisrooleja esimerkiksi projektityöskentelyä varten.

Hallittavien käyttöoikeuksien ja käyttöoikeustapahtumien määrä on olennainen roolien hallintaan vaikuttava tekijä. Laajoissa ympäristöissä rooleja tarvitaan enemmän, ja niiden hallitsemiseksi tarvitaan hierarkkisuutta, luokittelua, osaamista määrittelijöiltä sekä roolien ylläpitäjiltä. Tällöin on tarpeen myös tehdä koko käyttöoikeusprosessista hyvin standardoitu, automatisoitu ja välttää jouston mahdollisuutta. Liika jousto johtaa hallinnan katoamiseen.

Pienille ympäristöille, joissa hallitaan korkeintaan muutamia kymmeniä järjestelmiä ja muutamia tuhansia käyttäjiä pärjätään hyvin yksinkertaisella ja joustavalla mallilla. Pieniin ympäristöihin voidaan hyvin soveltaa 20/80-sääntöä, joka mukaan määritellään vain tärkeimmät roolit, joilla hallitaan 80% käyttöoikeuksista. Loput 20% käyttöoikeuksista voidaan hallita yksittäisinä ilman rooleja. Pieniin ympäristöihin sopivat joustavat roolit, jotka mahdollistavat dynaamisen prosessin aikana tapahtuvan roolien ja käyttöoikeuksien optimoinnin.

Vaikka roolien hallinnan avulla pyritään hallinnollisten töiden vähenemiseen ja prosessin nopeutumiseen, tulee huomioida myös vastuukysymykset. Yrityksen tietoturvapolitiikassa on määritettävä, mistä roolin käyttäjät, hyväksyjät ja roolien ylläpitäjät vastaavat. Tärkeää on määrittää, kuka ottaa vastuun roolista ja kuka roolin sisältämistä käyttöoikeuksista. Roolit on kehitetty tukemaan käyttöoikeuksien hallintaan liittyviä SOX-vaatimuksia, joiden mukaan käyttöoikeudet ja vastuut tulee pystyä jäljittämään tarvittaessa.

Roolien määrittämisen menetelmään vaikuttaa organisaation liiketoiminta- ja käyttöoikeusprosessien nykytila. Top-down-menetelmässä tutkitaan prosessikuvauksia, organisaatiorakenteita ja muodostetaan roolit niiden perusteella. Top-down-menetelmä vaatii hyvää liiketoiminnan ymmärtämistä. Bottom-up-menetelmässä kerätään tiedot tietojärjestelmien nykyisistä oikeuksista sekä tehdään henkilökohtaisia haastatteluja, joiden perusteella muodostetaan roolit. Bottom-up-menetelmä sopii, jos olemassa oleva käyttöoikeuspohja on kunnossa ja halutaan automatisoida määrittelyprosessia. Bottom-up-menetelmään on tarvittaessa saatavilla myös kaupallisia Role-mining-sovelluksia, jotka tulevat kysymykseen laajoissa järjestelmissä, joissa ”excel, kynä ja paperi” eivät riitä roolien selvittämiseen.

Määrittelyssä vaikea osa-alue on rajoitteiden huomioiminen. RBAC-standardissa kuvattuja rajoitteita ovat mm. kielletyt tai vaaralliset yhdistelmät ja sallittujen roolien lukumäärien rajoittaminen eri tilanteissa. Rajoitteita voidaan soveltaa samalla tavalla rooleille, istunnoille tai roolien sisältämille oikeuksille. Organisaatioiden vastuulle kuitenkin jää, kuinka rajoitteet määritellään ja mitä rajoitteita käytetään.

Roolien optimoinnin tavoitteena voidaan pitää määrällisesti ja laadullisesti oikean roolijoukon löytämistä. Tähän tavoitteeseen päästään noudattamalla vähäisimpien oikeuksien periaatetta, 20/80-sääntöä sekä oikein valittua roolien luokittelu- ja määrittelytapaa. Näitä periaatteita noudattamalla voidaan luoda sopiva joukko rooleja, kun organisaatiossa otetaan käyttöön roolipohjainen käyttöoikeuksien hallinta. Ensimmäisten roolien määrittämisen lisäksi yhtä tärkeää on hallita ylläpito ja roolisetin laajentaminen.

Järjestelmien osalta voidaan tehdä johtopäätös automatisoinnin tarpeesta: Mitä enemmän käyttöoikeustapahtumia, sitä suurempi tarve automatisoinnille. Automatiikka on myös tärkeä tekijä minimoitaessa käyttöoikeuksien saamisen läpimenoaikaa. Automatisointia ja integrointeja suunniteltaessa auttaa tarkastelu, jossa verrataan käyttöoikeuksien automatisoinnin tuomaa hyötyä investoinnin kustannuksiin. Joskus asioita kannattaa tehdä manuaalisestikin. Täysautomaattisen käyttöoikeusprosessin myötä roolit ja oikeudet tulevat kaikille ”kriteerit” täyttäville potentiaalisille henkilöille, vaikka tämä ei sovi kaikkiin tilanteisiin. Automatiikka tuo

nopeutta ja yksinkertaisuutta, mutta voi aiheuttaa enemmän työtä määrittelyssä ja aiheuttaa ylimääräisiä oikeuksia ja sitä myöten kasvattaa mm. lisenssikustannuksia.

Roolien hallinnan kannalta yksi suurimmista vaatimuksista järjestelmille on raportointi. Järjestelmistä tulisi olla tarkasti selvitettävissä, mitä rooleja kenellekin on myönnetty, milloin roolit on myönnetty ja mitä käyttöoikeuksia roolit tällä ajanhetkellä sisälsivät. Yhtä tarkasti tulisi olla tiedossa roolien muutokset, tekijät ja ajankohdat, jotta käyttöoikeuksien jäljitettävyyden periaate säilyy.

Sovelluskehittäjien tulisi huomioida roolipohjaisuus heti sovelluskehityksen alkuvaiheessa. Usein käyttöoikeuksien hallinnan toteutus jää sovelluskehityksen elinkaaren loppupäähän. Toteuttamiseen tulee käyttää yleisiä kehyksiä ja standardeja, jotta varmistetaan käyttöoikeuksien hallinnan ratkaisun tuleva elinkaari. Roolipohjaisen pääsynhallinnan kehityssuuntaus on selkeästi järjestelmien välinen yhteistoiminta sisältäen järjestelmäriippumattomien roolimääritysten käyttämisen.

Tärkein sovelluskehityksen standardi on RBAC INCITS 359-2004 , joka sisältää käsitemallin sekä funktiot roolien käsittelemiseen. Vuonna 2009 luonnosvaiheessa ollut standardi RBAC Implementation and Interoperability Standard (RIIS) sisältää mekanismit ja rajapinnat, joiden avulla RBAC-määrittelyt voidaan siirtää järjestelmästä toiseen. Palveluarkkitehtuuriin tarkoitettu standardi on XACML (eXtensible Access Control Markup Language), joka sisältää elementit RBAC-sovelluksen tekemiseen.

Tutkimusongelmana työssä kysyttiin, miksi teoria ei kohtaa reaalimaailmaa roolipohjaisessa käyttöoikeuksien hallinnassa. Roolien hallinnan standardeilla ja malleilla pyritään luomaan ideaalinen malli, jolla järjestelmiä voidaan hallita täysin roolien avulla. Roolien monipuoliset ominaisuudet mukaan lukien istuntojen hallinta, luvat, rajoitusten hallinta, hierarkkisuus ja erilaiset luokittelut mahdollistavat tämän periaatteen.

Standardien ja mallien heikkoutena voidaan pitää niiden keskittymistä yksittäisten järjestelmien käyttöoikeuksien hallintaan, vaikka niitä voidaan soveltaa myös työroolien hallintaan. Roolien avulla tulisi pysyä hallitsemaan organisaation kaikki käyttöoikeudet mukaan lukien järjestelmät, toimisto-ohjelmat, kulkuluvat ja kaikki asiat, joiden käyttämiseen tarvitaan lupa.

Reaalimaailmassa organisaation, järjestelmien ja prosessin keskeneräisyyden vuoksi tällainen ideaalinen malli on usein erittäin vaikeasti hallittava. Sääntöjen luominen koko ajan muuttuvaan ympäristöön on vaikeaa. Henkilöt ovat yksilöllisiä, mikä aiheuttaa vaikeuksia henkilöiden ryhmittelyyn rooleihin ja yhtenäisten toimintatapojen tekemiseen.

Onnistumisen edellytyksenä täytyy huomioida henkilöt ja henkilöiden osaamistaso. Roolien hallinta, määrittely ja ylläpito tehdään normaalisti henkilöiden toimesta, jotka eivät ole perehtyneet roolien hallintaan kovin syvällisesti. Tästä syystä roolien hallinnassa pitäisi pyrkiä yksikertaisuuteen ja toimintamalliin, joka on helposti ymmärrettävissä. Roolien määrän, luokittelun ja hierarkkisuuden minimointi helpottaa tässä. Jos osaamista ja asiaan perehtyneitä henkilöitä on enemmän, voidaan mennä monimutkaisempaan suuntaan.

Roolien hallinnan onnistuminen vaatii joustavat ja suoraviivaiset prosessit sekä niitä tukevat järjestelmät. Roolien mukaisessa käyttöoikeuksien hallinnassa ei ole tärkeintä tavoitella pelkästään ideaalista lopputulosta, vaan ideaalista lopputulosta oikeilla menetelmillä huomioiden toimintaympäristöön vaikuttavat tekijät.

Salon kaupungin käyttöoikeushallinnan projektin ensisijaisena tavoitteena oli saavuttaa hallittu käyttöoikeusprosessi. Käyttöoikeuksien hallintajärjestelmän käyttöönotto suoritettiin keväällä 2010. Järjestelmää käytettiin aluksi ilman rooleja, ja roolit haluttiin tuoda vasta myöhemmin osaksi käyttöoikeusprosessia. Projektissa luotiin täydet valmiudet roolien käytölle. Järjestelmä kehitettiin tukemaan Salon kaupungille sopivaa joustavaa mallia. Käyttäjämäärät ja tavoiteltu roolien määrä eivät vaatineet pitkälle vietyä automatiikkaa ja hierarkkisuutta. Roolien luokitteluksi riitti jako perusrooleihin ja tehtävärooleihin. Roolien avulla haluttiin hallita

suurinta osaa käyttöoikeuksista, ja loput oli tarkoitus hallita yksittäisinä käyttöoikeuksina.

Suomen terveydenhuollon lakien mukaan kaikilla julkisen terveydenhuollon palvelun antajilla on velvollisuus liittyä sähköisen kansallisarkiston palvelujen käyttäjiksi 1.4.2011 mennessä. Sähköisen arkiston vaatimuksiin kuuluu tietoturvallinen käyttöoikeuksien valvonta. Tämä tarkoittaa arkiston tietoja luovuttavan tai käyttävän henkilön ja henkilön roolin tunnistaminen. Osana tätä tietoturvapolitiikkaa käyttöoikeuksien hallinnan tulee tapahtua roolipohjaisesti, ja Salon kaupungilla on mahdollisuus tavoitteen täyttämiseen määräajan puitteissa.

Roolien käsittelyn osalta tarpeelliseksi kehityskohteiksi voi muodostua automatisoinnin laajentaminen ja roolien hierarkkisuuden tukeminen. Tämä on kuitenkin riippuvainen Salon kaupungin käyttämien roolien määrästä ja siitä, pystytäänkö rooleja hallitsemaan tarpeeksi hyvin nykyisillä periaatteilla. Jos roolien määrä pysyy pienenä, ei hierarkkisuudella saavuteta hyötyä. Jos pysytään tavoitelluissa yksinkertaisuus- ja 20/80-periaatteissa, niin suurta tarvetta järjestelmäkehitykselle ei ole. Tärkeintä on panostaa roolien määrittelyyn, jotta roolit tukevat mahdollisimman hyvin käyttöoikeuksien hallinnan kokonaisuutta.

Teoriaosuudessa esiteltiin skenaarioihin pohjautuva roolien määrittelyn malli, joka oli suunniteltu erityisesti Yhdysvaltojen terveydenhuollon tarpeisiin. Salon tapauksessa tämä malli olisi ollut liian raskas ja malli näyttäisi soveltuvan paremmin suuriin sairaalaympäristöihin. Yleisenä jatkokehityksenä olisi hyvä perehtyä tarkemmin HL7:n standardoimaan skenaariomalliin ja arvioida, toimisiko malli Suomen terveydenhuollon piirissä.

LÄHTEET

ANSI INCITS 359-2004. American National Standard for Information Technology, Role Based Access Control. [Viitattu 10.02.2010]. Saatavissa: http://www.cs.purdue.edu/homes/ninghui/readings/AccessControl/ANSI+INCITS+359-2004.pdf

Bednarz, A. 2005. Compliance: Thinking outside the Sarbox. Networkworld [Viitattu 10.02.2010]. Saatavissa: http://www.networkworld.com/research/2005/020705sox.html

Coyne, E. 2008. RBAC Implementation and Interoperability Standard (RIIS). INCITS CS1.1 RBAC Task Group [Viitattu 10.02.2010]. Saatavissa: http://csrc.nist.gov/groups/SNS/rbac/documents/incits-riis.pdf

GLB, Gramm–Leach–Bliley Act. 2010 [online]. Wikipedia [Viitattu 10.02.2010]. Saatavissa: http://en.wikipedia.org/wiki/Gramm-Leach-Bliley_Act

Gonzales-Webb, S. 2007. Implementing Role Based Access Control (RBAC) in Healthcare. VHA Veterans Health Administration [Viitattu 10.02.2010]. Saatavissa: http://www4.va.gov/rbac/docs/EHT_20070502_SW20_11_TEPR-RBAC_Presentation.pdf

Ferraiolo, D.F. & Kuhn, R. 1992. Role-Based Access Controls. National Institute of Standards and Technology, 15th National Computer Security Conference [Viitattu 10.02.2010]. Saatavissa: http://csrc.nist.gov/groups/SNS/rbac/documents/ferraiolo-kuhn-92.pdf

HIPAA, Health Insurance Portability and Accountability Act. 2010 [online]. Wikipedia [Viitattu 10.02.2010]. Saatavissa: http://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act

Hitachi ID Systems, Inc. Beyond Roles: A Practical Approach to Enterprise User Provisioning [online].[Viitattu 10.02.2010]. Saatavissa: http://www.idsynch.com/docs/beyond-roles.html

HL7 Security Technical Committee. 2005. HL7 Role-based Access Control (RBAC) Role Engineering Process. [Viitattu 10.02.2010]. Saatavissa: www.va.gov/rbac/docs/HIA_20051103_HL7_RBAC_Role_Engineering_Process_v1_1.doc

HL7 Standards, ANSI Approved Standards [online].[Viitattu 10.02.2010]. Saatavissa: http://www.hl7.org/implement/standards/ansiapproved.cfm

HL7 DSTU (Draft Standard for Trial Use) [online].[Viitattu 10.02.2010]. Saatavissa: http://www.hl7.org/dstucomments/

Jaideep, V., Vijayalakshmi, A. & Guo, Q. 2007. The Role Mining Problem: Finding a Minimal Descriptive Set of Roles. Rutgers University, Newark [Viitattu 10.02.2010]. Saatavissa: http://cimic.rutgers.edu/~jsvaidya/pub-papers/vaidya-sacmat07rm.pdf

KanTa – Kansallinen terveysarkisto [online]. [Viitattu 10.02.2010]. Saatavissa: https://www.kanta.fi/web/fi/kanta

Kern, A. 2002. Advanced Features for Enterprise-Wide Role-Based Access Control. Systor Security Solutions GmbH [Viitattu 10.02.2010]. Saatavissa: http://csrc.nist.gov/groups/SNS/rbac/standards.html

KPMG. 2004. Sarbanes-Oxley vaatii tehokkaampaa sisäistä valvontaa taloudelliseen raportointiin [online]. [Viitattu 10.02.2010]. Saatavissa: http://www.kpmg.fi/View041/sivu23.htm

Laki sosiaali- ja terveydenhuollon sähköisestä käsittelystä 159/2007. Annettu Helsingissä 9.2.2007. Saatavissa: http://www.finlex.fi/fi/laki/ajantasa/2007/20070159

Laki sähköisestä lääkemääräyksestä 61/2007. Annettu Helsingissä 2.2.2007. Saatavissa: http://www.finlex.fi/fi/laki/ajantasa/2007/20070061

Mienes, P. 2003. RBAC: Simply do it. KPMG [Viitattu 10.02.2010]. Saatavissa: http://www.bholdcompany.com/articles/bhold_RBAC_article_0.7b_EN.pdf

Mäkelä, N., 2008. Identiteetit ja roolit identiteetinhallintajärjestelmissä, Pro gradu –tutkielma. Tampere: Tampereen yliopisto, tietojenkäsittelytieteiden laitos

Neumann, G. & Strembeck, M. 2002. A ScenariodrivenRole Engineering Process for Functional RBAC Roles. proceedings of the 7th ACM Symposium on Access Control Models and Technologies [Viitattu 10.02.2010]. Saatavissa: http://csrc.nist.gov/groups/SNS/rbac/standards.html

Puustinen, A. 2008. Implementing Role Based Access Control into the Enterprise Environment to Support Business and IT Functions, Pro gradu -tutkielma. BHOLD [Viitattu 10.02.2010]. Saatavissa: http://www.bholdcompany.com/expertise/thesis/Masters%20Thesis_Puustinen_Antti.pdf

Propentus Oy, 2009. Roolipohjainen käyttöoikeuksien hallinta. Esitys.

RBAC & Sarbanes-Oxley Compliance [online]. National Institute of Standards and Technology [Viitattu 10.02.2010]. Saatavissa: http://csrc.nist.gov/groups/SNS/rbac/sarbanes_oxley.html

Roeckle, H., Schimpf, G. & Weidinger R. 2000. Process-Oriented Approach for Role-Finding to Implement Role-Based Security Administration in a Large Industrial Organization. [Viitattu 10.02.2010]. Saatavissa: http://www.roeckle.info/downloadables/acm-rbac2000-neu-05.pdf

Ruotsalainen, P. 2006. Suositukset terveydenhuollon asiakastietojen tietoturvalliselle sähköiselle arkistoinnille. STAKES raportteja 4 / 2006 [Viitattu 10.02.2010]. Saatavissa: http://www.stakes.fi/verkkojulkaisut/raportit/R4-2006-VERKKO.pdf

Sandhu, R.S., , Coynek,E.J., Feinsteink, H.L. & Youmank, C.E. 1996. Role-Based Access Control Models. IEEE Computer [Viitattu 10.02.2010]. Saatavissa: http://csrc.nist.gov/rbac/sandhu96.pdf

User Authorization with Role- Based Access Control. 2004. Veterans Health Administrator [Viitattu 10.02.2010]. Saatavissa: http://www4.va.gov/rbac/docs/20041013VHAPMIBrief.pdf

VM Valtiovarainministeriö, hallinnon kehittämisosasto. 2006. Käyttövaltuushallinnon hyvät periaatteet ja käytännöt.[Viitattu 10.02.2010]. Saatavissa: http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/05_valtionhallinnon_tietoturvallisuus/20061122Kaeyttoe/vahti_9_06.pdf

VM Valtiovarainministeriö, hallinnon kehittämisosasto. 2008. Valtionhallinnon tietoturvasanasto VAHTI[Viitattu 10.02.2010]. Saatavissa: http://www.vm.fi/vm/fi/13_hallinnon_kehittaminen/09_Tietoturvallisuus/02_tietoturvaohjeet_ja_maaraykset/index.jsp

LIITTEET

LIITE 1 (1/2). Tiivistelmä RBAC-standardin ANSI INCITS 359-2004 funktioista

A.1 Functional specification for Core RBAC

A.1.1 Administrative Functions

Creation and Maintenance of Element Sets: The basic element sets in Core RBAC are USERS, ROLES, OPS and OBS. Of these element sets, OPS and OBS are considered predefined by the underlying information system for which RBAC is deployed. For example, a banking system may have predefined transactions (OPS) for savings deposit and others, and predefined data sets (OBS) such as savings files, address files, and other necessary data. Administrators create and delete USERS and ROLES, and establish relationships between roles and existing operations and objects. Required administrative functions for USERS are AddUser and DeleteUser, and for ROLES are AddRole and DeleteRole.

Creation and Maintenance of Relations: The two main relations of Core RBAC are (a) user-to-role assignment relation (UA) and (b) permission-to-role assignment relation (PA). Functions to create and delete instances of User-to-Role Assignment (UA) relations are AssignUser and DeassignUser. For Permission-to-Role Assignment (PA) the required functions are GrantPermission and RevokePermission.

A.1.2 Supporting System Functions

• CreateSession: Creates a User Session and provides the user with a default set of active roles
• AddActiveRole: Adds a role as an active role for the current session
• DropActiveRole: Deletes a role from the active role set for the current session
• CheckAccess: Determines if the session subject has permission to perform the requested operation on an object.

A.1.3 Review Functions

• AssignedUsers (M): Returns the set of users assigned to a given role
• AssignedRoles (M): Returns the set of roles assigned to a given user
• RolePermissions (O): Returns the set of permissions granted to a given role
• UserPermissions (O): Returns the set of permissions a given user gets through his/her assigned roles
• SessionRoles(O): Returns the set of active roles associated with a session
• SessionPermissions (O): Returns the set of permissions available in the session (i.e., union of all permissions assigned to session's active roles)
• RoleOperationsOnObject (O): Returns the set of operations a given role may perform on a given object
• UserOperationsOnObject (O): Returns the set of operations a given user may perform on a given object (obtained either directly or through his/her assigned roles)

A.2 Functional specification for Hierarchical RBAC

A.2.1 Hierarchical Administrative Functions

• AddInheritance: Establish a new immediate inheritance relationship between two existing roles
• DeleteInheritance: Delete an existing immediate inheritance relationship between two roles
• AddAscendant: Create a new role and add it as an immediate ascendant of an existing role
• AddDescendant: Create a new role and add it as an immediate descendant of an existing role

A.2.2 Supporting System Functions

The Supporting System Functions for Hierarchical RBAC are the same as for Core RBAC and provide the same functionality. However, because of the presence of a role hierarchy, the functions CreateSession and AddActiveRole have to be redefined.

A.2.3 Review Functions

LIITE 1 (2/2). Tiivistelmä RBAC-standardin ANSI INCITS 359-2004 funktioista

• AuthorizedUsers: Returns the set of users directly assigned to a given role as well as those who were members of those “roles that inherited the given role”.
• AuthorizedRoles: Returns the set of roles directly assigned to a given user as well as those “roles that were inherited by the directly assigned roles”.
• RolePermissions: Returns the set of all permissions either directly granted to or inherited by a given role
• UserPermissions: Returns the set of permissions of a given user through his/her authorized roles (sum of directly assigned roles and roles inherited by those roles)
• RoleOperationsOnObject: Returns the set of operations a given role may perform on a given object (obtained either directly or by inheritance)
• UserOperationsOnObject: Returns the set of operations a given user may perform on a given object (obtained directly or through his/her assigned roles or through roles inherited by those roles)

A.3 Functional specification for SSD Relation

A.3.1 Administrative Functions

• CreateSSDSet: Create a named instance of an SSD relation
• DeleteSSDSet: Deletes an existing SSD relation
• AddSSDRoleMember: Adds a role to a named SSD role set
• DeleteSSDRoleMember: Deletes a role from a named SSD role set
• SetSSDCardinality: Sets the cardinality of the subset of roles from named SSD

A.3.2 Supporting System Functions

The Supporting System Functions for an SSD RBAC Model are the same as those for the Core RBAC Model.

A.3.3 Review Functions

• SSDRoleSets: Returns the set of named SSD relations created for the SSD RBAC model
• SSDRoleSetRoles: Returns the set of roles associated with a named SSD role set
• SSDRoleSetCardinality: Returns the cardinality of the subset within the named SSD role set for which common user membership restriction applies

A.4 Functional specification for DSD Relation

A.4.1 Administrative Functions

• CreateDSDSet: Create a named instance of DSD relation
• DeleteDSDSet: Deletes an existing DSD relation
• AddDSDRoleMember: Adds a role to a named DSD role set
• DeleteDSDRoleMember: Deletes a role from a named DSD role set
• SetDSDCardinality: Sets the cardinality of the subset of roles from named DSD role set for which user activation restriction within the same session applies

A.4.2 Supporting System Functions

• CreateSession: Creates a User Session and provides the user with a default set of active roles
• AddActiveRole: Adds a role as an active role for the current session
• DropActiveRole: Deletes a role from the active role set for the current session

A.4.3 Review Functions

• DSDRoleSets: Returns the set of named SSD relations created for the DSD RBAC model
• DSDRoleSetRoles: Returns the set of roles associated with a named DSD role set
• DSDRoleSetCardinality: Returns the cardinality of the subset within the named DSD role set for which user activation restriction within the same session applies

LIITE 2. RIIS-standardin funktiot

(Lähde: csrc.nist.gov/groups/SNS/rbac/documents/incits-riis.pdf)

Interaction Function	Meaning	Options
PostRoleSet	Inform of current set of roles	F, O, ULU, ULO
GetRoleSet	Obtain current set of roles	F, O, ULU, ULO
PostRoleName(rolename)	Inform of a new role name	F, O, ULU, ULO
GetRoleName(rolename)	Obtain new role name	F, O, ULU, ULO
PostUserSet	Inform of current set of RBAC users	F, O, ULU, ULO
GetUserSet	Obtain current set of RBAC users	F, O, ULU, ULO
PostRoleUsers(role name)	Inform of users currently assigned to a given role	F, O, ULU, ULO
GetRoleUsers(rolename)	Obtain users currently assigned to a given role	F, O, ULU, ULO
PostUserRoles(user)	Inform of roles currently assigned to a given user	F, O, ULU, ULO
GetUserRoles(user)	Obtain roles currently assigned to a given user	F, O, ULU, ULO
PostUserAssignment(user, role)	Inform of user assignment to a role	F, O, ULU, ULO
GetUserAssignment(user, role)	Obtain user assignment to a role	F, O, ULU, ULO
PostPermissionAssignment (role,permission)	Inform of permission assignment to a role	F, O, ULU, ULO
GetPermissionAssignment (role,permission)	Obtain permission assignment to a role	F, O, ULU, ULO
PostPermissionSet	Inform of current set of permissions	F, O, ULU, ULO
GetPermissionSet	Obtain current set of permissions	F, O, ULU, ULO

F – Fundamental O – Organizational
ULU – User Limiting-Universal ULO – User Limiting-Operational

22

Interaction Function	Meaning	Options
PostRolePermissions(role)	Inform of permissions currently assigned to a given role	F, O, ULU, ULO
GetRolePermissions(role)	Obtain permissions currently assigned to a given role	F, O, ULU, ULO
PostPermissionRoles (permission)	Inform of roles to which a given permission is assigned	F, O, ULU, ULO
GetPermissionRoles (permission)	Obtain roles to which a given permission is assigned	F, O, ULU, ULO
PostUserAssignmentConstraintStatic (user,role)	Inform of a given user's static assignment constraint	ULU
GetUserAssignmentConstraintStatic (user,role)	Obtain a given user's static assignment constraint	ULU
PostUserAssignmentConstraintDynamic (user,role)	Inform of a given user's dynamic assignment constraint	ULO
GetUserAssignmentConstraintDynamic (user,role)	Obtain a given user's dynamic assignment constraint	ULO
PostInheritanceRelationship (role,role)	Inform of an inheritance relationship between two given roles	O
GetInheritanceRelationship (role,role)	Obtain an inheritance relationship between two given roles	O

LIITE 3. Propentus Permission Manager -järjestelmän arkkitehtuuri